LIBRARY NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

NON-PARAMETRIC STATISTICAL SOFTWARE
FOR THE TRS-80 MICROCOMPUTER

by

Robert Lee Zangmeister, Jr.

December 1982

Thesis Advisors: J. D. Esary
R. R. Read

Approved for public release, distribution unlimited

T207861
T207861

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Non-Parametric Statistical Software for the TRS-80 Microcomputer

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
December 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Robert Lee Zangmeister, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, CA 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, CA 93940

12. REPORT DATE
December 1982

13. NUMBER OF PAGES
106

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Non-Parametric Statistical Software

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper documents the development of a non-parametric statistics package for the TRS-80 microcomputer. The package is comprised of ten programs with the major emphasis on non-parametric hypothesis testing.

The programming language is TRS-80 Level II Disk Basic. The package is compatible with any disk based TRS-80 Model I/III compatible microcomputer. With modification of the screen display and the disk access commands, the

package is transportable to microcomputers produced by other manufactures.

The statistical analysis capability implemented on a relatively inexpensive system provides a useful tool to the student or the trained analyst.

Approved for public release, distribution unlimited

Non-Parametric Statistical Software for the TRS-80 Microcomputer

by

Robert Lee Zangmeister, Jr.
B.E.E., University of Louisville, 1971

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN APPLIED SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December, 1982

TK2793
.A3
C.1

LIBRARY, NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93940

ABSTRACT

This paper documents the development of a non-parametric statistics package for the TRS-80 microcomputer. The package is comprised of ten programs with the major emphasis on non-parametric hypothesis testing.

The programming language is TRS-80 Level II Disk Basic. The package is compatible with any disk based TRS-80 Model I/III compatible microcomputer. With modification of the screen display and the disk access commands, the package is transportable to microcomputers produced by other manufacturers.

The statistical analysis capability implemented on a relatively inexpensive system provides a useful tool to the student or the trained analyst.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 7

II. OVERVIEW ---------- 9

A. PACKAGE CONTENTS ---------- 9

B. MICROCOMPUTER SYSTEM ---------- 10

C. LIMITATIONS ---------- 10

III. DISCUSSION OF PROGRAMS AND ALGORITHMS ---------- 12

A. STRUCTURE ---------- 12

B. PROGRAM 1, MENU/BAS ---------- 12

C. PROGRAM 2, DBMGR/BAS ---------- 14

D. PROGRAM 3, CTABLE/BAS ---------- 17

E. PROGRAM 4, WILCOX/BAS ---------- 18

F. PROGRAM 5, MANN/BAS ---------- 21

G. PROGRAM 6, SMIRNOV/BAS ---------- 24

H. PROGRAM 7, NLILL/BAS ---------- 27

I. PROGRAM 8, ELILL/BAS ---------- 30

J. PROGRAM 9, CHISQU/BAS ---------- 33

K. PROGRAM 10, NPSTAT/BAS ---------- 36

COMPUTER LISTINGS ---------- 44

LIST OF REFERENCES ---------- 103

BIBLIOGRAPHY ---------- 105

INITIAL DISTRIBUTION LIST ---------- 106

LIST OF TABLES

1. WILCOXSON DISTRIBUTION ACCURACY COMPARISON ----- 38

2. MANN-WHITNEY DISTRIBUTION ACCURACY COMPARISON -- 39

3. SMIRNOV DISTRIBUTION ACCURACY COMPARISON ------- 40

4. KOLMOGOROV DISTRIBUTION ACCURACY COMPARISON ---- 41

5. LILLIEFORS NORMALITY TEST DISTRIBUTION
ACCURACY COMPARISON ---------- 42

6. LILLIEFORS EXPONTETIALITY TEST DISTRIBUTION
ACCURACY COMPARISON ---------- 43

I. INTRODUCTION

Statistical Analysis of data is utilized as a decision aid in most every field. Personnel in technical fields have been exposed to and require the use of many statistical analysis procedures during the course of their career. In a career in which statistical analysis of data is required , but not on a daily basis, the knowledge of the proper statistical procedure is often retained where the mechanics of implementing the procedure requires extensive review. A software package which eliminates this review and handles the mechanics of implementing the procedure is most valuable.

The personal computer has evolved to the degree that it is now well suited for the implementation of statistical analysis procedures for small to moderately sized data bases. The statistical analysis software that is commercially available is generally unsophisticated and unsuited for the serious user.

An excellent package which models parametric distributions and implements parametric hypothesis testing was developed by Mr. R. P. Isbell and was the subject of his thesis work at the Naval Postgraduate School. The

purpose of this thesis is to provide a software package which implements the more commonly used non-parametric procedures, accurately approximates their distributions, and provides a convenient data base manager useful for both this package and that developed by Mr. Isbell.

II. OVERVIEW

A. PACKAGE CONTENTS

The package may be logically divided into three partitions; the data base manager, non-parametric distributions, and non-parametric hypothesis tests. The data base management programs permit data entry and storage to be in a columnar format or in a tabular format. The programs provide a convenient file editing capability as well as a flexible file printing capability.

The more commonly used non-parametric distributions and their inverse distributions are approximated. The distributions are:

1. Wilcoxson signed rank distribution
2. Mann-Whitney distribution
3. Smirnov distribution
4. Kolmogorov distribution
5. Lilliefors modification to the K-S test for Normal Distribution
6. Lilliefors modification to the K-S test for Exponential Distribution
7. Chi-Square distribution

The package contains procedures for implementing the following hypothesis tests:

1. Wilcoxson signed rank test
2. Mann-Whitney test
3. Smirnov test
4. Test for Normal Distribution
   a. Mean and variance unknown
   b. Mean known, variance unknown
   c. Mean and variance known
5. Test for Exponential Distribution

a. Parameter unknown
b. Parameter known

6. Contingency Table Chi-Square test

B. MICROCOMPUTER SYSTEM

The package was developed on an LNW Model I microcomputer system with 48K bytes of memory and one 5-1/4 inch floppy disk drive. The package is totally compatible with any TRS-80 Model I or Model III disk based microcomputer system and is not dependent on any unique features of the advanced TRS-80 Model I/III operating systems.

The programming language is TRS-80 Disk Basic implemented by Microsoft. Compatibility with all TRS-80 Model I/III systems is maintained by using only TRSDOS compatible commands. The package is not directly transportable to non-TRS-80 compatible computers; however, the Microsoft interpreter is commonly used by other manufacturers. The display commands would have to be altered to be compatible with microcomputers produced by other manufacturers.

C. LIMITATIONS

With any program development on a microcomputer, the programmer must be cognizant of its speed, memory, and precision limitations. To achieve maximum utilization of

the computer memory, the floppy disk is used as a virtual memory device pulling in routines on an as needed basis. This method permits a minimum of 3000 data observations to be resident in the computer memory.

The speed of execution is a major concern when using an interpreted language. To achieve an acceptable execution speed, asymptotic distributions were used in lieu of exact distributions and faster but more memory intensive sort routines were routines were used.

Precision is certainly a concern in this package; however, single precision variables proved to be adequate in most cases. Double precision variables were used only in the approximations to the Wilcoxson and the Mann-Whitney distributions.

III. DISCUSSION OF PROGRAMS AND ALGORITHMS

A. STRUCTURE

The package consists of ten programs, i.e. a control module, two data I/O programs, a program calculating distributions and six programs performing hypothesis tests. The choice of the program to be executed is made through the menu driven control module. The user is free to execute any program in the package through the control module with all variables being initialized as a new program is executed.

All data I/O is conducted through the two data I/O programs. The data is entered to the hypothesis test programs through a menu of established data files which were created by the data I/O programs.

B. PROGRAM 1, MENU/BAS

MENU/BAS is the control module from which all programs are executed. The program consists of the master menu and two sub-menus. The functional flow of the program is shown in figure 1.

BLOCK DIAGRAM OF MENU/BAS

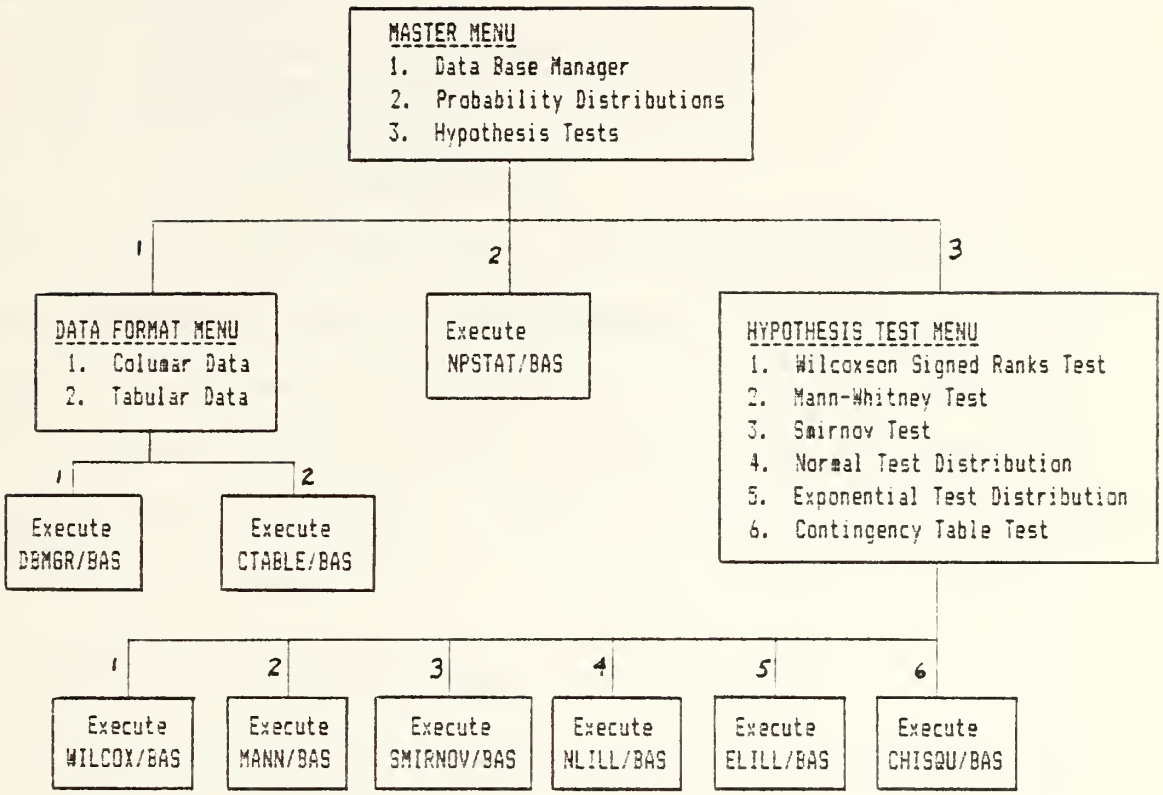

FIGURE 1.

C. PROGRAM 2, DBMGR/BAS

DBMGR/BAS provides the data I/O for the hypothesis test programs. The program is menu driven and offers the following selections:

1. Create New Data Files
2. Edit Data Files
3. Purge Data Files
4. Print Data Files
5. Return to Master Menu

A special purpose data file named EDITDATA/DIR is used by this program to contain the filenames of the data files created. It is used as a data file directory and is called by the hypothesis test programs to display the filenames of all current data files and to permit a convenient menu selection of the data files.

C.1. NEW FILE CREATION ROUTINE

The new file creation routine prompts the user for the number of files to be created. The user is then prompted for the disk drive number, i.e 0 - 3, to contain the files. The data files on the selected drive are displayed to prevent any undesired duplicate file naming. The user is then prompted for the new filenames. If a previously used filename is chosen, the user is alerted that a duplicate filename has been used and the user is given the option of renaming the file or overwriting the existing file.

After the files are named, the input screen is displayed. The screen will display four files (columns) and five observations (rows). The rows and columns may be scrolled to display other observations as needed.

The input area is located in the top-left section of the screen. Depressing ENTER transfers the input area to the bracketed destination area.

There are several special function keys which must be defined:

ENTER	-	transfers contents of the input area to the destination area. If the input area is blank, the destination area is moved to the right.
Arrows	-	moves the destination area
Shift arrow	-	moves the destination area to its most extreme position
D	-	deletes the current row
I	-	inserts a blank row immediately before the current row
CLEAR	-	Returns to menu
@	-	writes the files to disk and returns to the menu

The data is written to disk in a sequential file with the first field containing the number of observations and each field thereafter containing the observations. The routine was designed for maximum user convenience; however,

these convenience features will permit undesired "zero" valued observations if the user is not aware of the design of the program. Some precautions and guidelines to follow are:

1. All displayed files will be assigned a common value for the number of observations. Therefore, only files which will have the same number of observations should be entered simutaneously.

2. It is sometimes tempting to "play" with the arrow keys to move the destination area around the screen. The number of observations is determined by the maximum row reached by the destination area; therefore, "playing" with the arrow keys may result in an unwanted value for the number of observations.

C.2. EDIT EXISTING FILES ROUTINE

The file editing routine operates similiarly to the new file creation routine with only minor variation. The user is prompted for the disk drive containing the files and then prompted for the selection of the files to be edited. The files are displayed on the screen and editing is performed exactly as described in the new file creation routine. Prior to writing the files to disk, the user is given the opportunity to rename the file.

C.3. PURGE FILES

The data files are displayed one by one and the user is asked if it is desired to erase the file from the disk.

All erasing of data files should be performed from the purge routine and not from the operating system routines as the purge routine also removes the filename from the data file directory, EDITDATA/DIR.

C.4. PRINT DATA FILES

This routine permits formatting the data files for printing. The user is prompted for the files to be printed, the printer line width, column headers, and given the option of numbering the observations. The maximum number of spaces between columns is calculated and displayed. The user is prompted for the desired number of spaces between columns. Finally the user is asked for the report title and the files are printed.

D. PROGRAM 3, CTABLE/BAS

CTABLE/BAS is used to create tables used in the contingency table test program. The program operates similiarly to the DBMGR/BAS program with the differences listed below:

1. The data file names are stored in CTABLE/DIR versus EDITDATA/DIR.
2. In the new file creation routine, the user is prompted for the number of rows and columns in the table.
3. The "D" and "I" keys have no function.

E. PROGRAM 4, WILCOX/BAS

WILCOX/BAS performs the Wilcoxson Signed Rank test using data stored in data files created by DBMGR/BAS. The user is prompted for the two data files for the test. The test statistic is computed and the user is prompted for the hypothesis testing problem, i.e. one sided or two sided. The test statistic, sample size and p-value are displayed and the user is prompted for the significance level of the test. The significance level is compared with the p-value and acceptance or rejection of the null hypothesis is determined.

E.1. WILCOXSON SIGNED RANK TEST

The data consists of two paired samples of n observations, $(x_1,y_2),(x_2,y_2),..,(x_n,y_n)$. The difference $d_i = x_i - y_i$ is computed for each of the n pairs. Pairs with a difference of zero are omitted. The absolute differences, $|d_i|$, of the remaining pairs are ranked with rank 1 assigned to the pair (x_i,y_i) with the smallest absolute difference. If one or more pairs have the same absolute difference, the average rank is assigned to the pairs. The following test statistics are computed:

T^+ = Sum of ranks with positive d_i's

T^- = Sum of ranks with negative d_i's

$$T = MIN(T^+, T^-)$$

The assumptions for the test are:

1. The distribution of each d_i is symmetric
2. The pairs (x_i, y_i) constitute a bivariate random sample.

The hypothesis may be stated in one of three ways:

A. $H_o : E(X) = E(Y)$
   $H_1 : E(X) \neq E(Y)$

B. $H_o : E(X) \leq E(Y)$
   $H_1 : E(X) > E(Y)$

C. $H_o : E(X) \geq E(Y)$
   $H_o : E(X) < E(Y)$

The decision rule for the three hypotheses is to accept H_o if :

A. P(Test Statistic < T) > Significance Level/2

B. P(Test Statistic < T^+) > Significance Level

C. P(Test Statistic < T^-) > Significance Level

E.2. DISTRIBUTION THEORY AND ALGORITHM

The distribution of the Wilcoxson statistic, T, may be found by randomizing the signed ranks. Since there are two possible signs, + and −, and n ranks, the number of combinations in the randomization is 2^n. The exact cumulative distribution may be computed by examining the 2^n combinations and counting those with the sum of ranks

less than or equal to the test statistic and dividing by 2^n. Such a method is obviously slow for all but very small sample sizes.

A number of approximations to the exact distribution have been proposed. The most accurate of the proposed approximations is the Edgeworth approximation which is documented in References 1, 2, and 3. This approximation is based upon the fact that the distribution of T is asymptotically normal with mean and variance:

$$u_T = n(n+1)/4$$

$$\sigma_T^2 = n(n+1)(2n+1)/24$$

The normalized Wilcoxson Statistic, z, is defined:

$$z = (T - u_T)/\sigma_T$$

The Edgeworth expansion to terms of size $1/n^2$ is

$$P(T < t) = F(z) + L_4 f^{(3)}(z)/4! + L_6 f^{(5)}(z)/6! + 35(L_4)^2 f^{(7)}/8!$$

where F(z) is the cumulative standard normal distribution and :

$f^{(i)}$ is the i'th derivative of the normal density function

$L_4 = -12(3n^2+3n-1)/5n(n+1)(2n+1)$

$L_6 = 576(3n^4+6n^3-3n+1)/7(n(n+1)(2n+1))^2$

The algorithm implemented in this package uses a speed optimized exact distribution when the execution time is small and the Edgeworth approximation otherwise. The decision rule is :

Approximation	if	T > 27 or (N > 9 and T > N)
Exact	if	otherwise

The algorithm must use the normal cumulative distribution. The algorithm used in this package for the normal cumulative distribution was taken from the thesis work of Mr. R.P. Isbell and is documented in Reference 4.

F. PROGRAM 5, MANN/BAS

MANN/BAS performs the Mann-Whitney test using data stored in data files created by DBMGR/BAS. The user is prompted for the two data files for the test. The test statistic is computed and the user is prompted for the hypothesis testing problem, i.e. one-sided or two-sided. The test statistic, sample sizes and p-value are displayed and the user is prompted for the significance level of the test. The significance level is compared with the p-value and acceptance or rejection of the null hypothesis is determined.

F.1. MANN-WHITNEY TEST

The data consists of two random samples, $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$. The two samples are combined and ranked from 1 to n+m with the smallest observation being assigned the rank of 1. The test statistics are :

T_x = nm + n(n+1)/2 - (Sum of ranks for sample x)

T_y = nm + m(m+1)/2 - (Sum of ranks for sample y)

T = MIN(T_x, T_y)

The assumptions for the test are :

1. Both samples are random samples.
2. The two samples are independent.
3. If there is a difference between the two population distributions, that difference is in location only, i.e. $F_x(u) = F_y(u+c)$ where c is a constant.

The hypothesis may be stated in one of three ways :

A. H_o : $E(X) = E(Y)$
   H_1 : $E(X) \neq E(Y)$

B. H_o : $E(X) \leq E(Y)$
   H_1 : $E(X) > E(Y)$

C. H_o : $E(X) \geq E(Y)$
   H_1 : $E(X) < E(Y)$

The decision rule for the three hypotheses is to accept H_o if :

A. P(Test Statistic < T) > Significance Level/2

B. P(Test Statistic < T_x) > Significance Level

C. P(Test Statistic < T_y) > Significance Level

F.2. DISTRIBUTION THEORY AND ALGORITHM

The distribution of the Mann-Whitney statistic may be found by randomizing the ranks of the combined data. There are (n+m)!/(n!)(m!) ways in which the ranks may be grouped. The exact distribution may be computed by examining all possible combinations of ranks and counting those with the rank sum less than or equal to nm+n(n+1)/2 minus the test statistic and dividing by (n+m)!/(n!)(m!). Such a method is slow for all but very small sample sizes.

A number of approximations to the exact distribution have been proposed. The most accurate of those proposed is the Edgeworth approximation documented in Reference 5. The approximation is based on the fact that the test statistic is asymtotically normal with mean and variance :

$$u_t = nm/2$$

$$\sigma_t^2 = nm(n+m+1)/12$$

The normalized Mann-Whitney Statistic, z, is defined :

$$z = (T - u_t)/\sigma_t$$

The Edgeworth expansion is :

$$P(t<t) = F(z) + L_4 f^{(3)}(z)/4! + L_6 f^{(5)}(z)/6! + 35(L_4)^2 f_{(7)}(z)/8!$$

where F(z) is the standard normal cumulative distribution

$f^{(i)}$ is the i'th derivative of the standard normal density

$L^4 = -(m^2+n^2+n+m++nm)/(20nm(n+m+1))$

L^6 = NUM / DENOM

where :

$NUM = 2(n^4+m^4)+4nm(n^2+m^2)+6n^2m^2$

$+4(n^3+m^3)+7nm(n+m)+n^2+m^2+2nm-n-m$

$DENOM = 210n^2m^2(n+m+1)^2$

The algorithm that is implemented in this package uses a speed optimized exact distribution for small sample sizes and the Edgeworth approximation for larger sample sizes. The decision rule is :

approximation	if	m > 9 or T > 27 - n(n+1)/2
exact	if	otherwise

where n and m are the smaller and larger sample sizes respectively

G. PROGRAM 6, SMIRNOV/BAS

SMIRNOV/BAS performs the Smirnov test on data stored in data files by DBMGR/BAS. The user is prompted for the two data files for the test. The test statistic is computed and the user is prompted for the hypothesis testing problem, i.e. one-sided or two-sided. The test statistic, sample sizes, and p-value are displayed. The user is prompted for the significance level of the test.

The significance level of the test is compared to the p-value and acceptance or rejection of the null hypothesis is determined.

G.1. SMIRNOV TEST

The data consists of two independent random samples with unknown cumulative distribution functions F(x) and G(x). $S_1(x)$ and $S_2(x)$ are their empirical cumulative distribution functions. The test statistics are :

$$D^+ = MAX(S_1(x) - S_2(x))$$

$$D^- = MAX(S_2(x) - S_1(x))$$

$$D = MAX(D^+, D^-)$$

The assumptions for the test are :

1. The samples are random samples
2. The two samples are independent
3. The random samples are continuous

The hypotheses may be stated :

A. $H_o : F(x) = G(x)$
   $H_1 : F(x) \neq G(x)$

B. $H_o : F(x) \leq G(x)$
   $H_1 : F(x) > G(x)$

C. $H_o : F(x) \geq G(x)$
   $H_1 : F(x) < G(x)$

The decision rule for the three hypothese is to accept H_o if :

A. P(Test Statistic < D) > Significance Level/2

B. P(Test Statistic < D^+) > Significance Level

C. P(Test Statistic < D^-) > Significance Level

G.2. DISTRIBUTION THEORY

The exact distribution of the test statistic is difficult to analyze for the general case. E.F. Drion, in Reference 6, derived the distribution for the special case of equal sample sizes. It is known that the test statistic has the asymptotic distribution :

$$G(S) = 1 - \sum_{i} (-1)^{i-1} \exp(-2i^2 S^2)$$

where G(S) is known as the Smirnov distribution

S = D SQR((n+m)/nm) where n and m are sample sizes

It is generally agreed that sample sizes of 100 or more are required for the overestimating effects of the asymptotic distribution to become negligible. The problem is then to improve upon the Smirnov distribution for sample sizes of less than 100. P.J. Kim, in Reference 7, presented an improvement which reduced the overestimating effects for the smaller sample sizes. The improvements presented by P.J. Kim and implemented in this package are :

S = D•(SQR(nm/(n+m)))+2/(3•(SQR(m))) for m=kn, k=1,2,.....

S = (D-1/2m)•(SQR(n)) for n/m < 0.1

S = D•(SQR(nm/(n+m))) + 2/(5•(SQR(n))) otherwise

H. PROGRAM 7, NLILL/BAS

NLILL/BAS performs a test for normal distribution on a data set stored in a data file created by DBMGR/BAS. The test may be performed for the three cases :

1. mean and variance unknown
2. mean known and variance unknown
3. mean and variance known

The user is prompted for the data file to be tested and prompted for the specific case to be tested. In the cases where a parameter is known, the user is prompted for the known parameter. The test statistic, sample size and p-value are displayed. The user is prompted for the significance level of the test. The significance level is compared with the p-value and acceptance or rejection of the null hypothesis is determined.

H.1. TEST FOR NORMAL DISTRIBUTION

The data consists of a random sample $x_1,\ldots,x_n$ from an unknown cumulative distribution F(x). G(x) is the normal cumulative distribution with parameters either specified or estimated from the sample, depending upon the specific case. The test statistics are :

$$D^+ = MAX(i/n - G(x_i))$$

$$D^- = MAX(G(x_i) - (i-1)/n))$$

$$D = MAX(D^+, D^-)$$

The assumption for the test is that the sample is a random sample. The hypotheses are :

Case 1. H_o : F(x) is normal with unspecified mean and variance
H_1 : F(x) is not normal

Case 2. H_o : F(x) is normal with specified mean
H_1 : F(x) is not normal with specified mean

Case 3. H_o : F(x) is normal with specified mean and variance
H_1 : F(x) is not normal with specified mean and variance

The decision rule is accept H_o if :

P(Test Statistic < D) > Significance level

H.2. DISTRIBUTION THEORY AND ALGORITHM

The distribution of the test statistic differs for each of the three cases. For Case 1, where the mean and variance are unknown, the distribution was approximated using Monte Carlo techniques by H.W. Lilliefors and published in Reference 8. The distribution is presentd in tabular form and is not described in an analytical sense. In Reference 8, Lilliefors recognized that, for a particular significance level, the distribution of the test statistic differed by a multiplicative constant from that of the Kolmogorov distribution. This package capitalizes on this concept; however, the multiplicative value is a function of the value of the cumulative distribution. The solution must therefore be solved iteratively. The

algorithm follows:

```
T <--- Test Statistic
Denominator <--- 1
Do Until ABS(Tdenom - Denominator) < .0001
     D <--- T/Denominator
     S <--- D*(-.01 + SQR(N) + 0.85/SQR(N))
     Tdenom <--- Denominator
     CDF <--- 1 - (.135)^S
     If CDF < .85
        Then Denominator <--- 1 - .24xCDF
     If .85 < CDF < .90
        Then Denominator <--- 1.337 - .66xCDF
     If .90 < CDF < .95
        Then Denominator <--- 1.339 - .64xCDF
     If .95 < CDF < .975
        Then Denominator <--- 1.775 - 1.12xCDF
     If CDF > .975
        Then Denominator <--- 2.133 - 1.467xCDF
End Do
```

For Case 2, where the mean is known and the variance is unknown, the distribution was approximated by M.A. Stephens and published in Reference 9. A similiar solution was found to exist for this case as existed for Case 1, i.e. the cumulative distribution must be calculated iteratively. The algorithm for the iterative solution is :

```
T<---Test Stastistic
Denominator <--- 1
Do Until ABS(Tdenom - Denominator) < .0001
     D <--- T/Denominator
     S <--- D*(.35 + SQR(N) + .283/SQR(N))
     Tdenom <--- Denominator
     CDF <--- 1 - (.135)^S
     If CDF < .9
        Then Denominator <--- 1.83 - .8xCDF
     If .9 < CDF < .95
        Then Denominator <--- 1.47 - .4xCDF
     If .95 < CDF < .975
        Then Denominator <--- 1.85 - .8xCDF
     If CDF > .975
        Then Denominator <--- 1.07
End Do
```

For Case 3, where the mean and variance are completely specified, the distribution is the Kolmogorov distribution. The Kolmogorov distribution is a special case of the Smirnov distribution with one sample size approaching infinity and has the asymptotic distribution :

$$G(S) = 1 - \sum_{i} (-1)^{i-1} EXP(-2i^2 S^2)$$

with $S = D \cdot SQR(N)$

M.A. Stephens proposed improvements to the asymptotic distribution in Reference 9 which improve the accuracy of the asymptotic distribution for small sample sizes. This package implements the improvements proposed by Stephens in Reference 9, i.e. :

$S = D \cdot (.12 + SQR(n) + .11/SQR(n))$

I. PROGRAM 8, ELILL/BAS

ELILL/BAS performs a test for exponential distribution on data stored in data files by DBMGR/BAS. The test may be performed for the cases where the parameter is known and where the parameter is unknown.

The user is prompted for the data file to be tested and for the specific case, i.e. parameter known or unknown. In the case where the parameter is known, the user is prompted for the parameter value. The test statistic,

sample size and p-value are displayed. The user is prompted for the significance level of the test. The significance level is compared with the p-value and acceptance or rejection of the null hypothesis is determined.

I.1. TEST FOR EXPONENTIAL DISTRIBUTION

The data consists of a random sample $x_1,\ldots,x_n$ from some unknown cumulative distribution F(x). G(x) is the exponential cumulative distribution function with the parameter either estimated from the data or specified, depending upon the specifiic case. The test statistics are:

$$D^+ = MAX(i/n - G(x))$$

$$D^- = MAX(G(x) - (i-1)/n)$$

$$D = MAX(D^+, D^-)$$

The assumption for the test is that the sample is a random sample. The hypotheses are:

Case 1. H_o : F(x) is exponential - parameter specified
H_1 : F(x) is not exponential

Case 2. H_o : F(x) is exponential - parameter unspecified
H_1 : F(x) is not exponential

The decision rule is to accept H_o if :

P(Test Statistic < D) > Significance Level

I.2. DISTRIBUTION THEORY AND ALGORITHM

The distribution of the test statistic differs for the two cases. For the case where the parameter is unknown,

the distribution was approximated using Monte Carlo techniques by H.W. Lilliefors and published in Reference 10. M.A. Stephens, in Reference 9, expanded the work of Liffiefors by providing improvements for the small sample cases. The algorithm is an iterative method which assumes a solution and computes the test statistic which would yield the assumed value for the cumulative distribution function (CDF). The assumed CDF is updated until the difference between the calculated test statistic and the actual test statistic is sufficiently small. The algorithm is :

```
T <--- TEST STATISTIC
DENOMINATOR <--- .26 + SQR(n) + .5/SQR(N)
F1 <--- 0
F2 <--- 1
DO UNTIL ERROR < .0001
     CDF <--- (F2 - F1)/2
     IF CDF > .95
        THEN TS <--- 5.125xCDF - 3.7808
     IF .90 < CDF < .95
        THEN TS <--- 2.02xCDF - .8367
     IF .80 < CDF < .90
        THEN TS <--- 1.124xCDF - .0249
     IF .70 < CDF < .80
        THEN TS <--- 0.743xCDF + .2799
     IF .50 < CDF < .70
        THEN TS <--- 0.546xCDF + .4187
     IF .30 < CDF < .50
        THEN TS <--- 0.464xCDF + .4588
     IF .20 < CDF < .30
        THEN TS <--- 0.485xCDF + .4525
     IF .10 < CDF < .20
        THEN TS <--- 0.587xCDF + .4321
     IF CDF < .10
        THEN TS <--- 0.846xCDF + .4062
     TS <--- TS/DENOMINATOR + .2/N
     ERROR <--- T-TS
     IF ERROR > 0 THEN F1 <--- CDF ELSE F2 <--- CDF
END DO
```

For Case 2, with the parameter known, the distribution is the Kolmogorov distribution. The distribution and the algorithm used in this package is described in the fully specified case of the test for normal distribution.

J. PROGRAM 9, CHISQU/BAS

CHISQU/BAS performs the r x c contingency table test using data stored in data files created by CTABLE/BAS. The user is prompted for the data file to be tested. The test statistic, degrees of freedom, fraction of cells with expected frequency less than 5, number of cells with expected frequency less than 1, and the p-value are printed. The user is prompted for the significance level of the test. The significance level is compared with the p-value and acceptance or rejection of the null hypothesis is determined.

J.1. CONTINGENCY TABLE TEST

The contingency table test may be used to present a tabulation of data contained in several samples to test the hypothesis that the probabilities do not differ from sample to sample. Another use of the r x c contingency table is with a single sample where each element in the sample may be classified into one of r different categories according to one category and into one of c different categories according to a second category. The two applications are

treated the same in the statistical analysis, but basic differences between the two applications justify separate discussions.

J.1.1 TEST FOR DIFFERENCES IN CELL PROBABILITIES

The data consists of random samples drawn from r populations. Each observation in each sample may be classified into one of c different categories. Let O_{ij} be the number of observations from the i'th sample that fall into category j. Let n_i be the number of observations from the i'th population, c_j be the number of observations in the j'th category, and N be the total number of observations. The test statistic is :

$$T = \sum_i \sum_j (O_{ij} - E_{ij})^2 / E_{ij}$$

where $E_{ij} = n_i c_j / N$

The assumptions are that each sample is a random sample, the samples are mutually independent, and each observation may be categorized into exactly one of c categories. The hypotheses may be stated :

H_0 : All of the probabilities in the same column are equal

H_1 : At least two of the probabilities in the same column are not equal

The decision rule is to accept the null hypothesis if :

P(Test Statistic < T) > significance level

J.1.2 TEST FOR INDEPENDENCE

The data consists of one random sample of size N. The data may be classified by one of two criteria. Using the first criteria, each observation is associated with one of r rows and using the second criteria, each obsevatin is associatd with one of the c columns. Let O_{ij} be the number of observations in row i and column j, n_i be the number of observations in the i'th row, and c_j be the number of observations in the j'th column. The test statistic is as defined in the test for differences in cell probabilities.

The hypothesis may be stated :

H_0 : P(row i, column j) = P(row i) x P(column j)

H_1 : P(row i, column j) = P(row i) x P(column j)

The decision rule is to accept the null hypothesis if :

P(Test Statistic < T) > Significance Level

J.2 DISTRIBUTION THEORY AND ALGORITHM

The exact distribution of T is difficult to calculate for all but the 2 x 2 case. It is well known that T has asymptotically a Chi-Square distribution. The Chi-Square approximation is valid for large exected cell probabilities, i.e. E_{ij}. The approximation is

considered to be satisfactory if no E_{ij} is less than 1 and not more than 20% of the E_{ij}'s are less than 5.

The algorithm used in this package for the Chi-Square distribution was extracted from the thesis of Mr. R.P. Isbell. The details of his approximation to the distribution are detailed in Reference 1. For the 2 x 2 case, a correction factor proposed by F. Yates in Reference 11 is used to improve the Chi-Square approximation to the exact distribution.

K. PROGRAM 10, NPSTAT/BAS

NPSTAT/BAS contains seven non-parametric probability distributions, the normal distribution, and their inverse probability distributions. The distributions approximated are :

1. Normal Distribution
2. Wilcoxson Signed Rank Distribution
3. Mann-Whitney Distribution
4. Smirnov Distribution
5. Kolmogorov Distribution
6. Lilliefors Test for Normal Distribution
7. Lilliefors Test for Exponential Distribution

The user is provided a menu to select the distribution desired. The user is prompted for the required inputs and the cumulative distribution is displayed.

The approximations to the distributions are described in detail in the hypothesis test programs. The accuracy

comparisons of this packages approximations to the exact distributions are provided in Tables 1 through 6.

TABLE 1

WILCOXSON DISTRIBUTION ACCURACY COMPARISON

Sample Size	P≈0.05		P≈0.10		P≈0.20	
	Exact	Approx	Exact	Approx	Exact	Approx
5	.0313	.0313	.0938	.0938	.1563	.1563
10	.0420	.0420	.0962	.0970	.1875	.1875
15	.0473	.0473	.0938	.0938	.1947	.1947
20	.0487	.0487	.0947	.0947	.1942	.1942
25	.0479	.0479	.0954	.0954	.1942	.1942
30	.0481	.0481	.0990	.0990	.1965	.1965
35	.0484	.0484	.0977	.0977	.1972	.1970
40	.0486	.0486	.0984	.0984	.1987	.1987
45	.0498	.0498	.0981	.0981	.1973	.1973
50	.0495	.0495	.0988	.0988	.1994	.1994

TABLE 2

MANN-WHITNEY DISTRIBUTION ACCURACY COMPARISON

Sample Size	Sample Size	Test Statistic	Exact	Approx
6	6	7	.0465	.0468
6	6	10	.1201	.1200
6	6	12	.1970	.1965
6	8	11	.0539	.0542
6	8	14	.1142	.1143
6	8	16	.1725	.1724
8	8	16	.0524	.0524
8	8	20	.1172	.1173
8	8	22	.1641	.1641
8	10	21	.0506	.0506
8	10	25	.1015	.1016
8	10	28	.1577	.1577
10	10	28	.0526	.0526
10	10	33	.1088	.1088
10	10	36	.1575	.1575

TABLE 3

SMIRNOV DISTRIBUTION ACCURACY COMPARISON

Probablity = 0.95

	m = 25		m = 50		m = 100	
n/m	Exact	Approx	Exact	Approx	Exact	Approx
.1			.5209	.5185	.3797	.3737
.2	.5321	.5339	.3903	.3910	.2830	.2833
.3			.3393	.3434	.2443	.2463
.4	.4188	.4277	.3044	.3086	.2192	.2213
.5			.2769	.2765	.2003	.2003
.6	.3704	.3733	.2680	.2694	.1925	.1932
.7			.2579	.2571	.1848	.1844
.8	.3428	.3429	.2474	.2475	.1774	.1775
.9			.2409	.2397	.1726	.1719
1.0	.3087	.3082	.2256	.2258	.1634	.1635

Probablity = 0.975

	m = 25		m = 50		m = 100	
n/m	Exact	Approx	Exact	Approx	Exact	Approx
.1			.5775	.5743	.4224	.4142
.2	.5910	.5996	.4348	.4375	.3152	.3161
.3			.3777	.3829	.2719	.2742
.4	.4764	.4779	.3390	.3441	.2440	.2464
.5			.3096	.3094	.2236	.2235
.6	.4126	.4172	.2985	.3004	.2143	.2151
.7			.2871	.2867	.2056	.2053
.8	.3818	.3832	.2754	.2759	19744	.1976
.9			.2682	.2673	.1920	.1914
1.0	.3464	.3462	.2524	.2526	.1822	.1825

TABLE 4

KOLMOGOROV DISTRIBUTION ACCURACY COMPARISON

Sample	P = 0.90		P = 0.95		P = 0.975	
Size	Exact	Approx	Exact	Approx	Exact	Approx
5	.447	.446	.509	.509	.563	.564
10	.323	.323	.369	.369	.409	.409
15	.266	.267	.304	.304	.338	.338
20	.232	.232	.265	.265	.294	.294
25	.208	.209	.238	.238	.264	.264
30	.194	.191	.218	.218	.242	.242
35	.177	.177	.202	.202	.224	.224
40	.165	.166	.189	.189	.210	.210

TABLE 5

LILLIEFORS NORMALITY TEST DISTRIBUTION
ACCURACY COMPARISON

Sample Size	P = 0.90		P = 0.95		P = 0.99	
	Exact	Approx	Exact	Approx	Exact	Approx
5	.315	.314	.337	.343	.405	.396
10	.239	.239	.258	.261	.294	.302
15	.201	.200	.220	.219	.257	.253
20	.174	.176	.190	.192	.231	.222
25	.158	.159	.173	.173	.200	.200
30	.144	.146	.161	.159	.187	.184

TABLE 6

LILLIEFORS EXPONENTIALITY TEST DISTRIBUTION
ACCURACY COMPARISON

Sample Size	P = 0.80		P = 0.90		P = 0.95	
	Exact	Approx	Exact	Approx	Exact	Approx
5	.3603	.3615	.4045	.4028	.4420	.4400
10	.2626	.2642	.2955	.2956	.3244	.3239
15	.2174	.2185	.2448	.2448	.2690	.2686
20	.1893	.1905	.2132	.2137	.2345	.2346
25	.1703	.1711	.1918	.1921	.2110	.2110
30	.1559	.1567	.1756	.1760	.1932	.1933
35	.1447	.1454	.1630	.1633	.1793	.1795
40	.1356	.1362	.1528	.1531	.1681	.1683
45	.1281	.1286	.1443	.1445	.1588	.1589
50	.1217	.1221	.1371	.1373	.1509	.1510
60	.1118	.1117	.1255	.1256	.1381	.1381
70	.1033	.1035	.1164	.1164	.1281	.1281
80	.0968	.0969	.1090	.1091	.1200	.1200
90	.0914	.0914	.1029	.1029	.1132	.1132
100	.0868	.0868	.0977	.0977	.1075	.1075

PROGRAM 1 -- MENU/BAS

```
10 CLEAR 350
20 DEFINT I-K : DEFSTR Z
30 LB$=CHR$(123) : RB$=CHR$(125) : MB$=CHR$(176) :
BL$=CHR$(252)
40 CLS:PRINT TAB(10) "NON-PARAMETRIC STATISTICAL ANALYSIS
SOFTWARE"
50 PRINT TAB(31) "by"
60 PRINT TAB(21) "Robert Lee Zangmeister"
70 PRINT : PRINT TAB(13) "Submitted in partial fulfillment
of the" : PRINT TAB(17) "requirements for the degree of"
80 PRINT : PRINT TAB(19) "MASTER OF APPLIED SCIENCE"
90 PRINT : PRINT TAB(28) "from the"
100 PRINT : PRINT TAB(19) "NAVAL POSTGRADUATE SCHOOL"
110 PRINT TAB(25) "December 1982"
120 PRINT:PRINT TAB(17) "Press SPACE BAR to continue"
130 ZI=INKEY$ : IF ZI="" THEN 130
140 CLS : PRINT TAB(20);"PROGRAM SELECTION MENU"
150 PRINT @128,"(1)   Data Base Manager"
160 PRINT "(2)   Probability Distributions"
170 PRINT "(3)   Hypothesis Tests"
180 KP=896 : KS=6 : GOSUB 500
190 ON IO GOTO 200,250,1000
200 CLS:PRINT "DATA BASE OPTION":PRINT
210 PRINT "(1)   Column Format Data Entry"
220 PRINT "(2)   Table Format Data Entry"
230 KP=896:KS=2:GOSUB 500
240 IF IO = 1 THEN RUN "DBMGR/BAS" ELSE RUN "CTABLE/BAS"
250 RUN "NPSTAT/BAS"
500 REM  * * * KEYBOARD ENTRY ROUTINE * *
510 PRINT @KP+5,"OPTION DESIRED ";LB$;MB$;RB$;
520 ZI=INKEY$ : IF ZI="" THEN 520ELSE IO=VAL(ZI)
530 PRINT @KP+21, ZI;: FOR I=1 TO 100 : NEXT
540 IF IO<1 OR IO>KS THEN PRINT @KP, BL$;: PRINT @KP,"**
NOT A VALID OPTION   ***";: FOR I=1 TO 1000 : NEXT : PRINT
@KP, BL$; : GOTO 510
550 RETURN
1000 CLS:PRINT TAB(20);"HYPOTHESIS TEST MENU":PRINT
1010 PRINT @128,"(1)  Wilcoxson Signed Ranks Test"
1020 PRINT "(2)  Mann-Whitney Test"
1030 PRINT "(3)  Chi-Square Contingency Table Test"
1040 PRINT "(4)  Smirnov Test"
1050 PRINT "(5)  Test for Normality"
1060 PRINT "(6)  Test for Exponentiality"
1070 KP=896:KS=6:GOSUB 500
1080 ON IO GOTO 1100,1110,1120,1130,1140,1150
1100 RUN "WILCOX/BAS"
1110 RUN "MANN/BAS"
1120 RUN "CHISQU/BAS"
1130 RUN "SMIRNOV/BAS"
1140 RUN "NLILL/BAS"
1150 RUN "ELILL/BAS"
```

PROGRAM 2 - - - DBMGR/BAS

```
10 CLEAR 1000
20 ON ERROR GOTO 19000
30 DEFSTR Z
40 DIM F$(95),F1$(11),M$(11)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 CLS
70 PRINT TAB(20);"DATA FILE MANAGEMENT PROGRAM"
80 PRINT:PRINT:PRINT "MENU:"
90 PRINT "(1)  Create New Data Files"
100 PRINT "(2)  Edit Existing Files"
110 PRINT "(3)  Purge Existing Files"
120 PRINT "(4)  Print Data Files"
130 PRINT "(5)  Return to Master Menu"
140 KP=896:KS=5:GOSUB 180
150   ON IO GOSUB 260,310,330,340,170
160    GOTO 60
170 RUN "MENU/BAS"
180 REM * *  KEYBOARD INPUT SUBROUTINE *
190 PRINT @KP+5,"OPTION DESIRED ";LB$;MB$;RB$;
200 ZI=INKEY$:IF ZI="" THEN 200 ELSE IO=VAL(ZI)
210 PRINT @KP+21,ZI;:FOR I=1 TO 100:NEXT
220 IF IO<1 OR IO>KS THEN PRINT @KP,BL$;:PRINT @KP,"**
NOT A VALID OPTION  ***";: FOR I=1 TO 1000: NEXT: PRINT
@KP, BL$;:GOTO 190
230 RETURN
240 DN$=INKEY$:IF DN$="" THEN 240 ELSE I=VAL(DN$)
250 IF I<0 OR I>3 THEN 240 ELSE RETURN
260 'SUB TO  PERFORM NEW DATA FILE CREATION ROUTINES
270 CLS : INPUT "How Many Variables are to be Used
(1-11)";NF:CT=NF:RT=1
280 IF NF<1 OR NF>11 THEN 270
290 GOSUB 14000 : GOSUB 12000 : N=1 : CR=1 : GOSUB 6020
300 CLS:GOSUB 15050:RUN
310 GOSUB 350 : CR=1 : GOSUB 6020
320 GOSUB 15000 : RUN
330 GOSUB 5000 : RUN
340 GOSUB 350 : GOSUB 4000 : RUN
350 REM  * * *  READ MULTIPLE ARRAYS  * *
360 K=0 : DL$="/"
370 CLS:PRINT "Disk Drive Containing Files (0-3)?"
380 GOSUB 240
390 SL=1715:GOSUB 1200:IF SL=0 THEN GOTO 370
400 GOSUB 1400 : PRINT ""
410 PRINT:IF IO = 2 PRINT "Enter the Files To Be Edited
(e.g. 1/3/4)"; ELSE PRINT "Enter the Files To Be Printed";
420 INPUT B$ :P1=0
430 P2=INSTR(1,B$,DL$)
```

```
440 IF P2=0 THEN NV=VAL(B$): P1=1 : GOTO 460
450    NV=VAL(LEFT$(B$,P2-1)) : B$=RIGHT$(B$,LEN(B$)-P2)
460 IF NV>FO THEN PRINT:PRINT" * * * INVALID FILE # *
*":GOTO 410
470    K=K+1 : IF K=12 THEN NF=11 : GOTO 530
480    F1$(K)=F$(NV)
490    IF P1=0 THEN 430
500 B$="":PRINT:PRINT "Do You Wish To Read Files From
Another Disk(Y/N)? ";
510 ZI=INKEY$:IF ZI="" GOTO 510 ELSE IF ZI="Y" PRINT
ZI:GOTO 370
520 NF=K:CT=K:PRINT "N"
530 MM=-1
540 FOR I=1  TO NF
550 OPEN "I",1,F1$(I) : INPUT #1,M : CLOSE 1
560 IF M > MM THEN MM=M
570 NEXT I
580 RT=MM
590 GOSUB 12000 : PRINT
600 FOR J=1 TO NF
610     PRINT "READING IN FILE  --> ";F1$(J)
620     OPEN "I",1,F1$(J) : INPUT #1,N
630     FOR I=1 TO N : INPUT #1,X(I,J) : NEXT I
640     CLOSE 1
650 NEXT J
660 N=MM : RETURN
1000 REM  DIRECTORY CHECK  * *
1010 IF FO=0 THEN RETURN
1020 P2=0
1030 FOR P1=1 TO FO
1040     OPEN "I",1,F$(P1-P2)
1050     CLOSE 1
1060 NEXT P1
1070 IF P2=0 THEN RETURN ELSE GOSUB 3400 : RETURN
1100 REM * * *  CORRECT DIRECTORY  * *
1110 IF P1-P2=FO THEN 1130
1120    FOR I=P1-P2 TO FO-1 : F$(I)=F$(I+1) : NEXT I
1130 FO=FO-1 : P2=P2+1 : RETURN
1200 REM  * * *  READ DIRECTORY  * *
1210 CLS
1220 F$="EDITDATA/DIR:"+DN$
1230 OPEN "I",1,F$
1240 INPUT #1,FO
1250 IF FO=0 THEN 1290
1260 FOR I=1 TO FO
1270     INPUT #1,F$(I)
1280 NEXT I
1290 CLOSE 1 : GOSUB 1000
1300 CLOSE 1
1310 RETURN
1400 REM  * * *  DISPLAY DIRECTORY  * *
```

```
1410 CLS
1420 IF FO=0 THEN 1510
1430 FOR I=1 TO FO STEP 4
1440     FOR J=1 TO 4
1450         IF I+J-1>FO THEN 1510
1460         IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
1470       PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
1480     NEXT J
1490 NEXT I
1500 PRINT
1510 RETURN
2300 REM  * * *  CREATE DIRECTORY  * *
2305 CLOSE 1
2320 F$="EDITDATA/DIR:"+DN$
2340 OPEN "O",1,F$
2345 PRINT #1,O
2360 CLOSE 1
2370 RETURN
3400 REM  * * *  ENTER NEW FILES  * *
3410 F$="EDITDATA/DIR:"+DN$
3420 OPEN "O",1,F$
3430 PRINT #1,FO
3440 FOR I=1 TO FO
3450     PRINT #1,F$(I);",";
3460 NEXT I
3470 CLOSE 1
3480 RETURN
3500 REM  * * *  FILE NAME CHECK  * *
3510 EX=0
3520 FOR I=1 TO FO
3530     IF F$=F$(I) THEN EX=1 : RETURN
3540 NEXT I
3550 RETURN
3600 REM  * * *  DIRECTORY DISPLAY  * *
3610 CLS
3620 INPUT "DISK DRIVE NUMBER (0,1,2,3) TO FIND DIRECTORY
OF";DN$
3630 IF VAL(DN$)<0 OR VAL(DN$)>3 THEN 3610
3640 GOSUB 1200 : GOSUB 1400 : PRINT "" : INPUT "PRESS
ENTER TO CONTINUE";J1
3650 RETURN
4000 REM * * *  PRINT DATA FILES  * *
4005 CLS : INPUT "Input Printer Width";PW
4010 PF=0 : ML=-1 : DL$="!" : TL=-1
4020 FOR I=1 TO N
4030     FOR J=1 TO NF
4040         IF LEN(STR$(X(I,J))) > ML THEN
ML=LEN(STR$(X(I,J)))
4050     NEXT J
```

```
4060 NEXT I
4061 ML=ML+1
4070 CLS:PRINT:PRINT "Enter Column Headers.  Separating
lines with"
4080 PRINT"the delimiter '!' to produce a two column
header"
4100 FOR I=1 TO NF
4110 PRINT:PRINT "Enter the Column Header for Column";I
4120 INPUT M$(I)
4130 NEXT I
4140 MS=-1
4150 FOR I=1 TO NF
4160     P1=0
4170     P2=INSTR(P1+1,M$(I),DL$)
4180     L=P2-P1-1
4190     IF P2=0 THEN L=LEN(M$(I))-P1
4200     IF L>MS THEN MS=L
4210     IF P2=0 THEN 4240
4220        P1=P2
4230        GOTO 4170
4240 NEXT I
4250 IF ML>MS THEN MX=ML ELSE MX=MS
4260 FOR I=1 TO NF
4270     P1=0 : NL=0 : T$=""
4290     P2=INSTR(P1+1,M$(I),DL$)
4300     L=P2-P1-1
4310     IF P2=0 THEN L=LEN(M$(I))-P1
4320     SF=INT((MX-L)/2) : SB=MX-L-SF
4330     T$=T$+STRING$(SF,"
")+MID$(M$(I),P1+1,L)+STRING$(SB," ")
4335     NL=NL+1 : IF NL>TL THEN TL=NL
4340     IF P2=0 THEN M$(I)=T$ : GOTO 4370
4350        P1=P2
4360        GOTO 4290
4370 NEXT I
4380 PRINT:PRINT "Would you like to number the observations
(Y/N)? ";
4385 A$=INKEY$:IF A$="" THEN 4385 ELSE PRINT A$
4390 IF A$="Y" THEN PI=5 ELSE PI=0
4400 MS=INT((PW-MX*NF-PI)/NF)
4410 PRINT:PRINT"The maximum # of spaces that can be used
between columns is";MS;"."
4420 PRINT:INPUT"Enter the number of spaces between
columns";SP
4445 IF SP<0 THEN INPUT "TO RENAME COLUMNS TYPE Y ELSE
N";T$ : IF T$="Y" THEN 4010 ELSE RETURN
4450 IF SP<0 OR SP>MS THEN RETURN
4455 PRINT:INPUT "Enter Report Title";T$
4472 P1=(NF-1)*(MX+SP)+PI+MX-1
4474 P1=INT(P1/2) : P2=PW-P1 : IF LEN(T$)>PW THEN
T$=LEFT$(T$,PW)
```

```
4476 IF LEN(T$)>P2 THEN P1=INT((PW-LEN(T$))/2)
4480 LPRINT TAB(P1);T$
4490 LPRINT TAB(P1); : FOR I=1 TO LEN(T$) : LPRINT
"-";:NEXT I : LPRINT
4500 LPRINT "" : LPRINT ""
4505 FOR J=1 TO TL
4510     FOR I=1 TO NF
4520          LPRINT
TAB((I-1)*(MX+SP)+PI);MID$(M$(I),(J-1)X+1,MX);
4530     NEXT I
4540     LPRINT ""
4550 NEXT J
4560 A$="" : FOR I=1 TO MX : A$=A$+"-" : NEXT I
4570 FOR I=1 TO NF
4580     LPRINT TAB((I-1)*(MX+SP)+PI);A$;
4590 NEXT I
4600 LPRINT "" : K=PI+MX-1-INT((MX-ML)/2)
4610 FOR I=1 TO N
4620     IF PI=5 THEN LPRINT STR$(I);":";
4630     FOR J=1 TO NF
4640          LPRINT
TAB((J-1)*(MX+SP)-LEN(STR$(X(I,J)))+K);X(I,J);
4650     NEXT J
4660     LPRINT ""
4670 NEXT I
4680 RETURN
5000 REM  * * *  PURGE DATA FILES  * *
5010 CLS
5020 PRINT "Disk Drive Containing Files to Be Purged
(0-3)?"
5030 GOSUB 240
5040 SL=5010 : GOSUB 1200 : IF SL=0 THEN GOTO 5010
5045 K=0
5046 PRINT TAB(20);" ****** PURGE DATA FILES ****"
5050 FOR I=1 TO FO
5060 A$="":PRINT F$(I-K);"     KILL IT? (Y/N/Q) ";
5061 ZI=INKEY$:IF ZI="" THEN 5061
5062 IF ZI="Q" THEN PRINT ZI:GOTO 5095
5064 IF ZI="Y" THEN PRINT ZI:GOSUB 5100 ELSE PRINT "N"
5090 NEXT I
5095 GOSUB 3400
5099 RETURN
5100 KILL F$(I-K)
5110 IF I-K=FO THEN 5130
5120    FOR J=I-K TO FO-1 : F$(J)=F$(J+1) : NEXT J:
5130 FO=FO-1 : K=K+1
5140 RETURN
6000 REM *** EXAMINE DATA **
6005 INPUT CR
6010 IF CR < 1 THEN CR=1
6015 IF CR > N THEN CR=N
```

```
6020 E$=CHR$(31) : N$="" :P1=0: CLS
6025 C1=1:R1=1:B1=1
6030 CR=R1:CC=C1
6035 GOSUB 9500
6040 GOSUB 6300 : GOSUB 6350
6045 P1=0
6050 A$=INKEY$ : IF A$="" THEN 6050
6055 IF A$=CHR$(91) AND P1=0 THEN N$="": GOSUB 6500 : GOTO
6050
6060 IF A$=CHR$(27) AND P1=0 THEN N$="":CR=1:GOSUB
6520:GOTO 6050
6065 IF A$=CHR$(10) AND P1=0 THEN N$="": GOSUB 7000 : GOTO
6050
6070 IF A$=CHR$(26) AND P1=0 THEN N$="":CR=RT:GOSUB
7020:GOTO 6050
6075 IF A$=CHR$(8) AND P1=0  THEN N$="": GOSUB 7500 : GOTO
6050
6080 IF A$=CHR$(9) AND P1=0 THEN N$="": GOSUB 8000 : GOTO
6050
6085 IF A$=CHR$(24) AND P1=0 THEN N$="":CC=1:GOSUB
7530:GOTO 6050
6090 IF A$=CHR$(25) AND P1=0 THEN N$="":CC=CT:GOSUB
8030:GOTO 6050
6095 IF (A$>"/" AND A$<":")  OR A$="." OR A$="-" OR A$="+"
OR  A$="E" THEN P1=1 : GOSUB 9000 : GOTO 6050
6100 IF A$=CHR$(13) THEN P1=0: GOSUB 8500 : GOTO 6050
6105 IF A$=CHR$(8) AND P1=1 THEN GOSUB 9020 : GOTO 6050
6110 IF A$="D" AND P1=0 THEN N$="":GOSUB 6320:GOSUB
10000:GOTO 6050
6115 IF A$="I" AND P1=0 THEN N$="":GOSUB 6320:GOSUB
10500:GOTO 6050
6120 IF A$=CHR$(31) AND P1=0 THEN RUN
6125 IF A$="@" AND P1=0 THEN N=RT : RETURN
6130 GOTO 6050
6300 REM  * * *  CLEAR ENTRY AREA  * *
6310 CC=1
6320 PRINT @2,"              "
6330 PRINT @ 2,CHR$(95)
6340 RETURN
6350 REM  * * *  BRACKET ENTRY * *
6360 K=324+128*(CR-R1)+15*(CC-C1)
6370 TY=15360+K
6380 POKE TY,183 : POKE TY+14,187
6390 RETURN
6400 POKE TY,128 : POKE TY+14,128
6410 RETURN
6500 REM  * * *  PROCESS UP ARROW  * *
6510 CR=CR-1:IF CR<1 THEN CR=1:GOTO 6540
6520 IF CR<R1 THEN R1=CR:RL=R1+4:GOSUB 9500
6530 GOSUB 6400:GOSUB 6350
6540 RETURN
```

```
7000 REM  * * *  PROCESS DOWN ARROW  * *
7010 CR=CR+1
7020 IF CR>RL THEN RL=CR:R1=RL-4:GOSUB 6400:GOSUB 9500
7030 IF CR > RT THEN RT=RT+1
7040 GOSUB 6400:GOSUB 6350
7050 RETURN
7500 REM  * * *  PROCESS LEFT ARROW * *
7510 CC=CC-1
7520 IF CC<1 THEN CC=1:GOTO 7550
7530 IF CC<C1 THEN C1=CC:CL=C1+3:GOSUB 9500
7540 GOSUB 6400:GOSUB 6350
7550 RETURN
8000 REM  * * *  PROCESS RIGHT ARROW  * *
8010 CC=CC+1
8020 IF CC > CT THEN GOSUB 9200
8030 IF CC>CL THEN CL=CC:C1=CL-3:GOSUB 9500
8040 GOSUB 6400:GOSUB 6350
8050 RETURN
8500 REM  * * *  CHANGE VALUE IN FIELD  * *
8510 GOSUB 6320
8520 IF N$="" THEN GOSUB 8000 : RETURN
8530 IF CR>RT THEN RT=CR
8540 X(CR,CC)=VAL(N$)
8550 N$=""
8560 K=325+128*(CR-R1)+15*(CC-C1)
8570 PRINT @K+1,X(CR,CC);
8580 RETURN
8590 K=K-2:PRINT @K,CHR$(128);
8600 PRINT USING ZZ;X(CR,CC);
8610 RETURN
9000 REM * * *   PROCESS NUMBER ENTRY  * *
9010 N$=N$+A$ : PRINT @2,"           " : PRINT @ 2,N$;:PRINT
CHR$(95) : RETURN
9020 IF LEN(N$)>0 THEN N$=LEFT$(N$,LEN(N$)-1) : PRINT @ 2,"
          ": PRINT @ 2,N$
9030 RETURN
9200 REM  * * *  LAST COLUMN PROCESSING  * *
9210 CC=1 : C1 = CC : CL=C1+3
9220 CR=CR+1
9230 IF CR>RL THEN RL=CR : R1=RL-4
9240 GOSUB 6400
9250 IF CT > 4 OR CR = RL THEN GOSUB 9500
9260 IF CR>RT THEN RT=RT+1
9270 RETURN
9500 REM * * *  SCREEN DISPLAY  * *
9510 PRINT @196,CHR$(143) : IF CT<C1+4 THEN CL=CT ELSE
CL=C1+3
9520 RL=R1+4
9530 FOR I=C1-1 TO CL-1
9540 PRINT @197+15*(I+1-C1),STRING$(6,CHR$(143));:PRINT
CHR$(128);:PRINT CHR$(65+I);:PRINT CHR$(128);:PRINT
```

```
STRING$(6,CHR$(143))
9550 NEXT I
9560 PRINT @195,CHR$(170):PRINT @259,CHR$(170)
9570 FOR I=R1-1 TO RL-1
9580 IF I > 8 THEN K1=0 ELSE K1=1
9590 PRINT @320+128*(I+1-R1)+K1,STR$(I+1);:PRINT CHR$(170)
9600 FOR J=C1-1 TO CL-1
9610 PRINT @326+15*(J+1-C1)+128*(I+1-R1),X(I+1,J+1)
9620 NEXT J
9630 PRINT @387+128*(I+1-R1),CHR$(170)
9640 NEXT I
9650 RETURN
10000 REM  * * *  DELETE A LINE  * *
10010 FOR I=CR TO RT-1:FOR J=1 TO CT:X(I,J)=X(I+1,J) : NEXT
J : NEXT I
10020 FOR J=1 TO CT: X(RT,J)=0 : NEXT J
10030 RT=RT-1 : IF R1 > 1 AND RL > RT THEN R1=R1-1 :
CR=CR-1
10040 GOSUB 9500 : GOSUB 6350
10050 RETURN
10500 REM  * * *  INSERT A LINE  * *
10510 FOR I=RT TO CT STEP -1 : FOR J=1 TO CT :
X(I+1,J)=X(I,J) : NEXT J : NEXT I
10520 FOR J=1 TO CT : X(CR,J)=0 : NEXT J
10530 RT=RT+1 : GOSUB 9500 : GOSUB 6350
10540 RETURN
11000 REM  * * *  VERIFY FILENAME  * *
11010 FW=0
11020 P1=INSTR(1,S$,"/")
11030 IF P1 <> 0 THEN IF LEN(S$)-P1>3 THEN FW=1 : RETURN
11040 IF P1 >8 THEN FW=1 : RETURN
11050 IF P1=0 THEN IF LEN(S$)>8 THEN FW=1 : RETURN
11060 IF P1 <> 0 THEN IF ASC(MID$(S$,P1+1,1))<65 OR
ASC(MID$(S$,P1+1,1))>90 THEN FW=1 : RETURN
11070 IF ASC(LEFT$(S$,1))<65 OR ASC(LEFT$(S$,1))>90 THEN
FW=1 : RETURN
11080 IF P1=0 THEN P2=LEN(S$) ELSE P2=P1-1
11090 P3=1
11100 FOR I=P3 TO P2
11110       A$=MID$(S$,I,1)
11120       IF (ASC(A$)>47 AND ASC(A$)<58) OR (ASC(A$)>64
AND ASC(A$)<91 ) THEN 11140
11130          FW=1 : RETURN
11140    NEXT I
11150 IF P3=1 AND P1 <> 0 THEN P3=P1+1 : P2=LEN(S$) : GOTO
11100
11160 RETURN
12000 REM  * * *  DIMENSION ARRAYS * *
12010 NV=MEM-500 : NV=INT(NV/(4*(NF+1)))-1
12020 IF MM > NV THEN PRINT "INSUFFICIENT SPACE TO READ IN.
":INPUT "PRESS ENTER TO CONTINUE";A$
```

```
12030 DIM X(NV,NF)
12040 MN=NV
12050 RETURN
13000 REM  * * *   RENAME FILES ROUTINE  * *
13010 CLS
13020 PRINT "Disk Drive To Contain Files (0-3)?"
13030 GOSUB 240
13040 SL=8240:GOSUB 1200 : GOSUB 1400 :  PRINT ""
13050 FOR J=1 TO NF
13060     PRINT "Variable #";J;"Currently Uses Filename
-->";F1$(J)
13070     INPUT "New Name";S$ : GOSUB 11000
13080     IF FW=1 THEN PRINT "INVALID FILENAME" : GOTO
13060
13090        F$=S$ : GOSUB 3500
13100       IF EX=1 THEN PRINT:PRINT "A File Currently Uses
the New Filename.":PRINT "Do You Want to Use it Anyway
(Y/N)?" ELSE GOTO 13120
13110 A$=INKEY$:IF A$="" THEN 13110 ELSE PRINT A$: IF A$<>
"Y" THEN 13060
13120        F1$(J)=F$
13130 NEXT J
13140 RETURN
14000 REM  * * *  NAME FILES ROUTINE  * *
14010 CLS
14020 PRINT "Disk Drive To Contain Files (0-3)?":GOSUB 240
14030 SL=8330 : GOSUB 1200 : GOSUB 1400 : PRINT ""
14040 PRINT
14050 FOR J=1 TO NF
14060 PRINT "FILENAME FOR VARIABLE #";J : INPUT S$ : GOSUB
11000
14070 IF FW=1 THEN PRINT "INVALID FILENAME" : GOTO 14060
14080    F$=S$ : GOSUB 3500
14090    IF EX=1 THEN PRINT "FILENAME IS ALREADY USED. TYPE
Y TO USE NAME ANYWAY ELSE N";: INPUT A$ : IF A$ <> "Y" THEN
14060
14100    F1$(J)=F$
14110 NEXT J
14120 RETURN
15000 REM  * * *  EXITING ROUTINE FOR OPTIONS 1 & 2  * *
15010 CLS
15020 PRINT "Do You Wish To Rename Files (Y/N)?"
15030 ZI=INKEY$:IF ZI="" GOTO 15030
15040 IF ZI="Y" THEN GOSUB 13000
15050 SL=8550 : GOSUB  1200 : GOSUB 1400 : PRINT ""
15060 PRINT:PRINT
15070 FOR  K=1 TO NF
15080        PRINT "WRITING  TO DISK --> ";F1$(K)
15090      OPEN "O",1,F1$(K)+":"+DN$
15100      PRINT #1,N : FOR I=1 TO N : PRINT #1,X(I,K); :
NEXT I
```

```
15110       CLOSE 1 : F$=F1$(K) : GOSUB 3500 : IF EX=1 THEN
15130
15120       FO=FO+1 :  F$(FO)=F1$(K) : GOSUB 3400
15130 NEXT K
15140 RETURN
19000 'ERROR HANDLING ROUTINE
19015 IF ERR=10 AND ERL=8520 THEN N$="" : RESUME 8530
19020 IF ERR <> 106 THEN 19100
19021 IF ERL=5100 THEN RESUME 5110
19022 IF ERL=1040 THEN GOSUB 1100 : RESUME 1050
19025 IF ERL <> 1230 THEN 19100
19030 IF SL=1715 OR SL=5010 OR SL=5410  THEN PRINT "NO
FILES ON THIS DRIVE. PRESS ENTER TO CONTINUE": INPUT A$ :
SL=0 : RESUME 1300
19045 IF SLIsA8240 OR SL=8330  OR SL=8550 THEN GOSUB 2300 :
RESUME 1210
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0 :
RESUME
```

PROGRAM 3, CTABLE/BAS

```
10 CLEAR 1000
20 ON ERROR GOTO 19000
30 DEFSTR Z
40 DIM F$(95),M$(11),N$(11)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 Z$="######":ZZ="##.##[[[["
70 CLS
80 PRINT TAB(20);"DATA FILE MANAGEMENT PROGRAM"
90 PRINT:PRINT:PRINT "MENU:"
100 PRINT "(1)  Create New Data Files"
110 PRINT "(2)  Edit Existing Files"
120 PRINT "(3)  Purge Existing Files"
130 PRINT "(4)  Print Data Files"
140 PRINT "(5)  Return to Master Menu"
150 KP=896:KS=5:GOSUB 190
160   ON IO GOSUB 1000,1100,1200,1300,180
170    GOTO 70
180 RUN "MENU/BAS"
190 REM * *  KEYBOARD INPUT SUBROUTINE *
200 PRINT @KP+5,"OPTION DESIRED ";LB$;MB$;RB$;
210 ZI=INKEY$:IF ZI="" THEN 210 ELSE IO=VAL(ZI)
220 PRINT @KP+21,ZI;:FOR I=1 TO 100:NEXT
230 IF IO<1 OR IO>KS THEN PRINT @KP,BL$;:PRINT @KP,"**
NOT A VALID OPTION  ***";: FOR I=1 TO 1000: NEXT: PRINT
@KP, BL$;:GOTO 200
240 RETURN
250 DN$=INKEY$:IF DN$="" THEN 250 ELSE I=VAL(DN$)
260 IF I<0 OR I>3 THEN 250 ELSE RETURN
1000 REM * * *  NEW FILE ROUTINE * *
1010 CLS : INPUT "How many rows";RT
1020 INPUT "How many columns";CT
1030 BT=1
1040 GOSUB 14000 : GOSUB 12000 : N=1 : CR=1 : GOSUB 6040
1050 CLS:GOSUB 15050:RUN
1100 GOSUB 1500 : CR=1 : GOSUB 6040
1110 GOSUB 15000 : RUN
1200 GOSUB 5000 : RUN
1300 GOSUB 1500 : GOSUB 4000 : RUN
1500 REM * * * READ DATA ARRAY * *
1510  DL$="/"
1520 CLS:PRINT "Disk Drive Containing Files (0-3)?"
1530 GOSUB 250
1540 SL=1715:GOSUB 2400:IF SL=0 THEN GOTO 1520
1550 GOSUB 2800 : PRINT ""
1560 PRINT:IF IO = 2 PRINT "Enter the File To Be Edited";
ELSE PRINT "Enter the File To Be Printed";
```

```
1570 INPUT NV
1580 IF NV>FO THEN PRINT:PRINT" * * * INVALID FILE # *
*":GOTO 1560
1590    F1$=F$(NV)
1600 OPEN "I",1,F1$
1610 INPUT #1,RT
1620 INPUT #1,CT
1630 INPUT #1,BT
1640 CLOSE 1
1650 GOSUB 12000 : PRINT
1660 OPEN "I",1,F1$
1670 INPUT #1,RT:INPUT #1,CT:INPUT #1,BT
1680 FOR I=1 TO RT
1690 FOR J=1 TO CT
1700 INPUT #1,X(I,J)
1710 NEXT J
1720 NEXT I
1730 CLOSE 1
1740 RETURN
2000 REM * * * CHECK FOR FILE DIRECTORY * *
2010 IF FO=0 THEN RETURN
2020 P2=0
2030 FOR P1=1 TO FO
2040     OPEN "I",1,F$(P1-P2)
2050     CLOSE 1
2060 NEXT P1
2070 IF P2=0 THEN RETURN ELSE GOSUB 3400 : RETURN
2080 'SUB TO CORRECT DIRECTORY
2090 IF P1-P2=FO THEN 2110
2100    FOR I=P1-P2 TO FO-1 : F$(I)=F$(I+1) : NEXT I
2110 FO=FO-1 : P2=P2+1 : RETURN
2400 REM * * * READ DIRECTORY OF FILES * *
2410 CLS
2420 F$="CTABLE/DIR:"+DN$
2430 OPEN "I",1,F$
2440 INPUT #1,FO
2450 IF FO=0 THEN 2490
2460 FOR I=1 TO FO
2470     INPUT #1,F$(I)
2480 NEXT I
2490 CLOSE 1 : GOSUB 2000
2500 CLOSE 1
2510 RETURN
2800 REM  * * *  DISPLAY DIRECTORY  * *
2810 CLS
2820 IF FO=0 THEN 2910
2830 FOR I=1 TO FO STEP 4
2840     FOR J=1 TO 4
2850         IF I+J-1>FO THEN 2910
2860         IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
```

```
2870         PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
2880      NEXT J
2890 NEXT I
2900 PRINT
2910 RETURN
3200 REM  * * * CREATE "CTABLE/DIR" * *
3210 DN$="O"
3220 CLOSE 1
3230 F$="CTABLE/DIR:"+DN$
3240 OPEN "O",1,F$
3250 PRINT #1,O
3260 CLOSE 1
3270 RETURN
3400 REM  * * *  WRITE NEW FILES IN DIRECTORY  * *
3410 F$="CTABLE/DIR:"+DN$
3420 OPEN "O",1,F$
3430 PRINT #1,FO
3440 FOR I=1 TO FO
3450      PRINT #1,F$(I);",";
3460 NEXT I
3470 CLOSE 1
3480 RETURN
3600 REM * * * CHECK IF FILENAME WAS ALREADY USED * *
3610 EX=0
3620 FOR I=1 TO FO
3630      IF F$=F$(I) THEN EX=1 : RETURN
3640 NEXT I
3650 RETURN
4000 REM * * * PRINT ROUTINE * *
4010 CLS : INPUT "Input Printer Width";PW
4020 PF=0 : ML=-1 : DL$="!" : TL=-1
4030 FOR I=1 TO N
4040      FOR J=1 TO CT
4050           IF LEN(STR$(X(I,J))) > ML THEN
ML=LEN(STR$(X(I,J)))
4060      NEXT J
4070 NEXT I
4080 ML=ML+1
4090 CLS:PRINT:PRINT "Enter Column Headers.  Separating
lines with"
4100 PRINT"the delimiter '!' to produce a two column
header"
4110 FOR I=1 TO CT
4120 PRINT:PRINT "Enter the Column Header for Column";I
4130 INPUT M$(I)
4140 NEXT I
4150 MS=-1
4160 FOR I=1 TO CT
4170      P1=0
4180      P2=INSTR(P1+1,M$(I),DL$)
```

```
4190      L=P2-P1-1
4200      IF P2=0 THEN L=LEN(M$(I))-P1
4210      IF L>MS THEN MS=L
4220      IF P2=0 THEN 4250
4230         P1=P2
4240         GOTO 4180
4250 NEXT I
4260 IF ML>MS THEN MX=ML ELSE MX=MS
4270 FOR I=1 TO CT
4280      P1=0 : NL=0 : T$=""
4290      P2=INSTR(P1+1,M$(I),DL$)
4300      L=P2-P1-1
4310      IF P2=0 THEN L=LEN(M$(I))-P1
4320      SF=INT((MX-L)/2) : SB=MX-L-SF
4330      T$=T$+STRING$(SF,"
")+MID$(M$(I),P1+1,L)+STRING$(SB," ")
4340      NL=NL+1 : IF NL>TL THEN TL=NL
4350      IF P2=0 THEN M$(I)=T$ : GOTO 4380
4360         P1=P2
4370         GOTO 4290
4380 NEXT I
4390 PRINT:PRINT "Would you like to number the rows (Y/N)?
";
4400 A$=INKEY$:IF A$="" THEN 4400 ELSE PRINT A$
4410 IF A$="Y" THEN PI=5 ELSE PI=0
4420 MS=INT((PW-MX * CT-PI)/CT)
4430 PRINT:PRINT"The maximum # of spaces that can be used
between columns is";MS;"."
4440 PRINT:INPUT"Enter the number of spaces between
columns";SP
4450 IF SP<0 THEN INPUT "TO RENAME COLUMNS TYPE Y ELSE
N";T$ : IF T$="Y" THEN 4020 ELSE RETURN
4460 IF SP<0 OR SP>MS THEN RETURN
4470 PRINT:INPUT "Enter Report Title";T$
4480 P1=(CT-1)*(MX+SP)+PI+MX-1
4490 P1=INT(P1/2) : P2=PW-P1 : IF LEN(T$)>PW THEN
T$=LEFT$(T$,PW)
4500 IF LEN(T$)>P2 THEN P1=INT((PW-LEN(T$))/2)
4510 LPRINT TAB(P1);T$
4520 LPRINT TAB(P1); : FOR I=1 TO LEN(T$) : LPRINT
"-";:NEXT I : LPRINT
4530 LPRINT "" : LPRINT ""
4540 FOR J=1 TO TL
4550      FOR I=1 TO CT
4560           LPRINT
TAB((I-1)*(MX+SP)+PI);MID$(M$(I),(J-1)X+1,MX);
4570      NEXT I
4580      LPRINT ""
4590 NEXT J
4600 A$="" : FOR I=1 TO MX : A$=A$+"-" : NEXT I
4610 FOR I=1 TO CT
```

```
4620      LPRINT TAB((I-1)*(MX+SP)+PI);A$;
4630 NEXT I
4640 LPRINT "" : K=PI+MX-1-INT((MX-ML)/2)
4650 FOR I=1 TO RT
4660      IF PI=5 THEN LPRINT STR$(I);":";
4670      FOR J=1 TO CT
4680           LPRINT
TAB((J-1)*(MX+SP)-LEN(STR$(X(I,J)))+K);X(I,J);
4690      NEXT J
4700      LPRINT ""
4710 NEXT I
4720 RETURN
5000 REM  * * *  PURGE DATA FILE DIRECTORY * *
5010 CLS
5020 PRINT "Disk Drive Containing Files to Be Purged
(0-3)?"
5030 GOSUB 250
5040 SL=5010 : GOSUB 2400 : IF SL=0 THEN GOTO 5010
5050 K=0
5060 PRINT TAB(20);" ****** PURGE DATA FILES ****"
5070 FOR I=1 TO F0
5080 A$="":PRINT F$(I-K);"     KILL IT? (Y/N/Q) ";
5090 ZI=INKEY$:IF ZI="" THEN 5090
5100 IF ZI="Q" THEN PRINT ZI:GOTO 5130
5110 IF ZI="Y" THEN PRINT ZI:GOSUB 5150 ELSE PRINT "N"
5120 NEXT I
5130 GOSUB 3400
5140 RETURN
5150 KILL F$(I-K)
5160 IF I-K=F0 THEN 5180
5170    FOR J=I-K TO F0-1 : F$(J)=F$(J+1) : NEXT J:
5180 F0=F0-1 : K=K+1
5190 RETURN
6000 REM *** EXAMINE DATA **
6010 INPUT CR
6020 IF CR < 1 THEN CR=1
6030 IF CR > N THEN CR=N
6040 E$=CHR$(31) : N$="" :P1=0: CLS
6050 C1=1:R1=1:B1=1
6060 CR=R1:CC=C1
6070 GOSUB 6760
6080 GOSUB 6240 : GOSUB 6290
6090 P1=0
6100 A$=INKEY$ : IF A$="" THEN 6100
6110 IF A$=CHR$(91) AND P1=0 THEN N$="": GOSUB 6360 : GOTO
6100
6120 IF A$=CHR$(27) AND P1=0 THEN N$="":CR=1:GOSUB
6380:GOTO 6100
6130 IF A$=CHR$(10) AND P1=0 THEN N$="": GOSUB 6410 : GOTO
6100
6140 IF A$=CHR$(26) AND P1=0 THEN N$="":CR=RT:GOSUB
```

```
6440:GOTO 6100
6150 IF A$=CHR$(8) AND P1=0  THEN N$="": GOSUB 6470 : GOTO
6100
6160 IF A$=CHR$(9) AND P1=0 THEN N$="": GOSUB 6530 : GOTO
6100
6170 IF A$=CHR$(24) AND P1=0 THEN N$="":CC=1:GOSUB
6500:GOTO 6100
6180 IF A$=CHR$(25) AND P1=0 THEN N$="":CC=CT:GOSUB
6560:GOTO 6100
6190 IF (A$>"/" AND A$<":")  OR A$="E" THEN P1=1 : GOSUB
6720 : GOTO 6100
6200 IF A$=CHR$(13) THEN P1=0: GOSUB 6590 : GOTO 6100
6210 IF A$=CHR$(8) AND P1=1  THEN GOSUB 6740 : GOTO 6100
6220 IF A$="@" AND P1=0 THEN RETURN
6225 IF A$=CHR$(31) AND P1=0 THEN RUN
6230 GOTO 6100
6240 REM  * * *  CLEAR ENTRY AREA  * *
6250 CC=1
6260 PRINT @2,"             "
6270 PRINT @ 2,CHR$(95)
6280 RETURN
6290 REM  * * *  BRACKET ENTRY * *
6300 K=320+128*(CR-R1)+11*(CC-C1)+5
6310 TY=15360+K
6320 POKE TY,183 : POKE TY+11,187
6330 RETURN
6340 POKE TY,128 : POKE TY+11,128
6350 RETURN
6360 REM  * * *  PROCESS UP ARROW  * *
6370 CR=CR-1:IF CR<1 THEN CR=1:GOTO 6400
6380 IF CR<R1 THEN R1=CR:RL=R1+4:GOSUB 6760
6390 GOSUB 6340:GOSUB 6290
6400 RETURN
6410 REM  * * *  PROCESS DOWN ARROW  * *
6420 CR=CR+1
6430 IF CR>RT THEN CR=RT:GOTO 6460
6440 IF CR>RL THEN RL=CR:R1=RL-4:GOSUB 6340:GOSUB 6760
6450 GOSUB 6340:GOSUB 6290
6460 RETURN
6470 REM  * * *  PROCESS LEFT ARROW * *
6480 CC=CC-1
6490 IF CC<1 THEN CC=1:GOTO 6520
6500 IF CC<C1 THEN C1=CC:CL=C1+4:GOSUB 6760
6510 GOSUB 6340:GOSUB 6290
6520 RETURN
6530 REM  * * *  PROCESS RIGHT ARROW  * *
6540 CC=CC+1
6550 IF CC > CT THEN GOSUB 7000
6560 IF CC>CL THEN CL=CC:C1=CL-4:GOSUB 6760
6570 GOSUB 6340:GOSUB 6290
6580 RETURN
```

```
6590 REM  * * *  CHANGE VALUE IN FIELD  * *
6600 GOSUB 6260
6610 IF N$="" THEN GOSUB 6530 : RETURN
6620 X(CR,CC)=VAL(N$)
6630 N$=""
6640 K=326+128*(CR-R1)+11*(CC-C1)
6650 IF X(CR,CC)>999999 THEN GOTO 6690
6660 PRINT @K+1,CHR$(128);
6670 PRINT USING Z$;X(CR,CC);
6680 RETURN
6690 K=K-2:PRINT @K,CHR$(128);
6700 PRINT USING ZZ;X(CR,CC);
6710 RETURN
6720 REM * * *   PROCESS NUMBER ENTRY  * *
6730 N$=N$+A$ : PRINT @2,"           " : PRINT @ 2,N$;:PRINT
CHR$(95) : RETURN
6740 IF LEN(N$)>0 THEN N$=LEFT$(N$,LEN(N$)-1) : PRINT @ 2,"
          ": PRINT @ 2,N$
6750 RETURN
6760 PRINT @196,CHR$(143) : IF CT<C1+5 THEN CL=CT ELSE
CL=C1+4
6770 IF RT<R1+4 THEN RL = RT ELSE RL=R1+4
6780 FOR I=C1-1 TO CL-1
6790 PRINT @197+11*(I+1-C1),STRING$(4,CHR$(143));:PRINT
CHR$(128);:PRINT CHR$(65+I);:PRINT CHR$(128);:PRINT
STRING$(4,CHR$(143))
6800 NEXT I
6810 PRINT @195,CHR$(170):PRINT @259,CHR$(170)
6820 FOR I=R1-1 TO RL-1
6830 IF I > 8 THEN K1=0 ELSE K1=1
6840 PRINT @320+128*(I+1-R1)+K1,STR$(I+1);:PRINT CHR$(170)
6850 FOR J=C1-1 TO CL-1
6860 PRINT @327+11*(J+1-C1)+128*(I+1-R1),CHR$(128);:PRINT
USINGZ$;X(I+1,J+1)
6870 NEXT J
6880 PRINT @387+128*(I+1-R1),CHR$(170)
6890 NEXT I
6900 RETURN
7000 REM * * *  PROCESS LAST COLUMN ENTRY  * *
7010 CC=1 : C1=CC : CL=C1+4
7020 CR=CR+1 : IF CR > RT THEN CR=RT
7030 IF CR > RL THEN RL=CR : R1=RL-4
7040 GOSUB 6340
7050 IF CT > 5 OR CR = RL THEN GOSUB 6760
7060 RETURN
11000 REM * * * VERIFY FILENAME - FW=1 IF FAILS * *
11010 FW=0
11020 P1=INSTR(1,S$,"/")
11030 IF P1 <> 0 THEN IF LEN(S$)-P1>3 THEN FW=1 : RETURN
11040 IF P1 >8 THEN FW=1 : RETURN
11050 IF P1=0 THEN IF LEN(S$)>8 THEN FW=1 : RETURN
```

```
11060 IF P1 <> 0 THEN IF ASC(MID$(S$,P1+1,1))<65 OR
ASC(MID$(S$,P1+1,1))>90 THEN FW=1 : RETURN
11070 IF ASC(LEFT$(S$,1))<65 OR ASC(LEFT$(S$,1))>90 THEN
FW=1 : RETURN
11080 IF P1=0 THEN P2=LEN(S$) ELSE P2=P1-1
11090 P3=1
11100 FOR I=P3 TO P2
11110       A$=MID$(S$,I,1)
11120       IF (ASC(A$)>47 AND ASC(A$)<58) OR (ASC(A$)>64
AND ASC(A$)<91 ) THEN 11140
11130          FW=1 : RETURN
11140    NEXT I
11150 IF P3=1 AND P1 <> 0 THEN P3=P1+1 : P2=LEN(S$) : GOTO
11100
11160 RETURN
12000 REM * * * DIMENSION ARRAYS * *
12010 NV=MEM-500 : NV=INT(NV/(4*(RT * CT * BT+1)))-1
12020 IF MM > NV THEN PRINT "INSUFFICIENT SPACE TO READ IN.
":INPUT "PRESS ENTER TO CONTINUE";A$
12030 DIM X(RT,CT)
12040 MN=NV
12050 RETURN
13000 REM * * *   RENAME FILES    * *
13010 CLS
13020 PRINT "Disk Drive To Contain Files (0-3)?"
13030 GOSUB 250
13040 SL=8240:GOSUB 2400 : GOSUB 2800 :  PRINT ""
13050      INPUT "New Name";S$ : GOSUB 11000
13060      IF FW=1 THEN PRINT "INVALID FILENAME" : GOTO
13050
13070         F$=S$ : GOSUB 3600
13080        IF EX=1 THEN PRINT:PRINT "A File Currently Uses
the New Filename.":PRINT "Do You Want to Use it Anyway
(Y/N)?" ELSE GOTO 13100
13090 A$=INKEY$:IF A$="" THEN 13090 ELSE PRINT A$: IF A$<>
"Y" THEN 13050
13100         F1$=F$
13110 RETURN
14000 REM * * *   NAME FILE     * *
14010 CLS
14020 PRINT "Disk Drive To Contain File (0-3)?":GOSUB 250
14030 SL=8330 : GOSUB 2400 : GOSUB 2800 : PRINT ""
14040 PRINT
14050 PRINT "FILENAME FOR TABLE"; : INPUT S$ : GOSUB 11000
14060 IF FW=1 THEN PRINT "INVALID FILENAME" : GOTO 14050
14070     F$=S$ : GOSUB 3600
14080     IF EX=1 THEN PRINT "FILENAME IS ALREADY USED. TYPE
Y TO USE NAME ANYWAY ELSE N";: INPUT A$ : IF A$ <> "Y" THEN
14050
14090     F1$=F$
14100 RETURN
```

```
15000 REM * * * EXITING ROUTINE FOR OPTIONS 1 & 2 * *
15010 CLS
15020 PRINT "Do You Wish To Rename File (Y/N)?"
15030 ZI=INKEY$:IF ZI="" GOTO 15030
15040 IF ZI="Y" THEN GOSUB 13000
15050 SL=8550 : GOSUB  2400 : GOSUB 2800 : PRINT ""
15060 PRINT:PRINT
15070       PRINT "WRITING  TO DISK --> ";F1$
15080       OPEN "O",1,F1$+":"+DN$
15090 PRINT #1,RT:PRINT #1,CT:PRINT#1,BT
15100 FOR I=1 TO RT
15110 FOR J=1 TO CT
15120 PRINT #1,X(I,J);
15130 NEXT J
15140 NEXT I
15150       CLOSE 1 : F$=F1$ : GOSUB 3600 : IF EX=1 THEN
15170
15160       FO=FO+1 :  F$(FO)=F1$ : GOSUB 3400
15170 RETURN
19000 REM * * * ERROR HANDLING ROUTINE * *
19010 IF ERR=10 AND ERL=6590 THEN N$="" : RESUME 6630
19020 IF ERR <> 106 THEN 19080
19030 IF ERL=5150 THEN RESUME 5160
19040 IF ERL=2040 THEN GOSUB 2080 : RESUME 2050
19050 IF ERL <> 2430 THEN 19080
19060 IF SL=1715 OR SL=5010 OR SL=5410  THEN PRINT "NO
FILES ON THIS DRIVE. PRESS ENTER TO CONTINUE": INPUT A$ :
SL=0 : RESUME 2500
19070 IF SL=8240 OR SL=8330  OR SL=8550 THEN GOSUB 3200 :
RESUME 2410
19080 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0 :
RESUME
```

PROGRAM 4, WILCOX/BAS

```
10 CLEAR 1000
20 ON ERROR GOTO 19000
30 DEFINT P:DEFSTR Z:DEFDBL F,L
40 DIM S(1500,1),F$(95),P(1500),X(1500),B(50)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 REM * * *  WILCOXSON SIGNED RANKS TEST  * *
70 CLS:PRINT @20,"WILCOXSON SIGNED RANKS TEST"
75 PRINT:PRINT:PRINT "Disk Drive Containing Data Files
(0-3)?"
80 DN$=INKEY$:IF DN$="" THEN 80 ELSE I=VAL(DN$)
85 IF I<0 OR I>3 THEN 75
90 SL=100 : GOSUB 11000 : IF SL=0 THEN 70 ELSE GOSUB 12000
: PRINT ""
100 PRINT:INPUT "Enter the Two Data Files (e.g. 3,5)";F1,F2
110 IF F1>F0 OR F2>F0 OR F1<1 OR F2<1 THEN 100
120 OPEN "I",1,F$(F1)+":"+DN$
130 INPUT #1,N
140 FOR I=1 TO N : INPUT #1,S(I,0) : NEXT I
150 CLOSE 1
160 OPEN "I",1,F$(F2)+":"+DN$
170 INPUT #1,M
180 FOR I=1 TO M : INPUT #1,S(I,1) : NEXT I
190 CLOSE 1
200 IF N=M GOTO 250
210 PRINT:PRINT "The Data Sets are NOT of Equal Size.  Do
You Want to Continue using the Smaller # of data points,
Enter other Data Files, or Return to the Master Menu
(C/E/R)?";
220 ZI=INKEY$:IF ZI="" THEN 220
230 IF ZI="C" THEN PRINT ZI:GOTO 240:ELSE IF ZI="E" THEN
PRINT "E":RUN ELSE IF ZI"R" THEN PRINT "R":RUN"MENU/BAS"
ELSE GOTO 220
240 IF N > M THEN N=M
250 J=1
260 FOR I=1 TO N
270     S(J,1)=S(I,0)-S(I,1)
280     IF S(J,1)=0 THEN J=J-1
290     J=J+1
300 NEXT I
310 N=J-1
320 FOR I=1 TO N : X(I)=ABS(S(I,1)) : NEXT I
330 M=1 : N1=N : FOR I=1 TO N1  : P(I)=I : NEXT I : GOSUB
10000
340 JS=1
350 FOR K=2 TO N
360     R=ABS(X(P(JS))/X(P(K)))
```

```
370      IF R < .999 OR R > 1.001 THEN JE=K-1 : GOSUB 14000
: JS=K
380 NEXT K
390 JE=N : GOSUB 14000
400 TP=0:TN=0
410 FOR I=1 TO N
420      IF S(I,1) > 0 THEN TP=TP+S(I,1) ELSE
TN=TN+ABS(S(I,1))
430 NEXT I
440 IF TP > TN THEN TS=TN ELSE TS=TP
450 CLS:PRINT@20,"WILCOXSON SIGNED RANKS TEST":PRINT:PRINT
460 PRINT "HYPOTHESIS          NULL
ALTERNATIVE"
470 PRINT "    (1)         E(X)  =  E(Y)              E(X) =/=
E(Y)"
480 PRINT "    (2)         E(X) <=  E(Y)              E(X)  >
E(Y)"
490 PRINT "    (3)         E(X) =>  E(Y)              E(X)  <
E(Y)"
500 PRINT:PRINT "Where X and Y represent Data Sets 1 & 2
Respectively"
510 JP=640:KS=3:GOSUB 8000
520 IF IO=1 V=TS ELSE IF IO=2 V=TP ELSE IF IO=3 V=TN
530 CLS:PRINT@20,"WILCOXSON SIGNED RANKS TEST":PRINT:PRINT
540 PRINT "Test Statistic = ";V
545 PRINT "Sample Size = ";N
550 GOSUB 2000
560 ON IO GOSUB 600,800,1000
570 PRINT @960,"Press SPACE BAR when ready to return to
main menu";
580 IF INKEY$= "" GOTO 580 ELSE RUN "MENU/BAS"
600 REM * * * NULL HYPOTHESIS #1 * *
610 PRINT "P-Value = ";:PRINT USING "#.####";2 * FOX
620 PRINT:INPUT"Enter Desired Alpha Level";AL
630 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 490
640 PRINT:PRINT "For Alpha =";AL;", ";
650 IF AL<2 * FOX THEN PRINT "accept"; ELSE PRINT "reject";
660 PRINT " the hypothesis that E(X) = E(Y)"
670 PRINT "-vs- the alternative that E(X) =/= E(Y)."
680 RETURN
800 REM * * * NULL HYPOTHESIS #2 * *
810 PRINT "P-Value = ";:PRINT USING "#.####";FOX
820 PRINT:INPUT"Enter Desired Alpha Level";AL
830 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 620
840 PRINT:PRINT "For Alpha =";AL;", ";
850 IF AL<FOX THEN PRINT "accept"; ELSE PRINT "reject";
860 PRINT " the hypothesis that E(X) <= E(Y)"
870 PRINT "-vs- the alternative that E(X) > E(Y)."
880 RETURN
```

```
1000 REM * * * NULL HYPOTHESIS #3 * *
1010 PRINT "P-Value = ";:PRINT USING "#.####";FOX
1020 PRINT:INPUT"Enter Desired Alpha Level";AL
1030 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A
VLAUE BETWEEN 0 & 1":GOTO 020
1040 PRINT:PRINT "For Alpha =";AL;", ";
1050 IF AL<FOX THEN PRINT "accept"; ELSE PRINT "reject";
1060 PRINT " the hypothesis that E(X) => E(Y)"
1070 PRINT "-vs- the alternative that E(X) < E(Y)."
1080 RETURN
2000 REM * * *  WILCOXSON DISTRIBUTION * *
2020 PX=V:IN=N:FLAG=0
2030 IF V>IN*(IN+1)/2 OR V<0 OR IN<1 THEN PRINT " * * ERROR
* *":GOTO 100
2040 IF V>IN*(IN+1)/4 THEN V=IN*(IN+1)/2-V-1:FLAG=1
2050 IF V>27 OR (IN>9 AND V>IN) THEN GOSUB 2600 ELSE GOSUB
2070
2060 RETURN
2070 IF INT(V)=V GOSUB 2110 ELSE GOTO 2090
2080 RETURN
2090 V=INT(V):GOSUB 2110:TFOX=FOX:V=V+1:GOSUB
2110:FOX=(FOX+TFOX)/2:QOX=1-FOX
2100 RETURN
2110 I1(2)=0:I1(3)=0:I1(4)=0:I1(5)=0:I1(6)=0
2120 C=1
2130 N=INT(.5*(-1+SQR(1+8*V)))
2140 FOR I=1 TO N
2150 I1(1)=0
2160 I1(1)=I1(1)+1
2170 IF I<2 GOTO 2320
2180 I1(2)=I1(1)
2190 I1(2)=I1(2)+1
2200 IF I<3 GOTO 2320
2210 I1(3)=I1(2)
2220 I1(3)=I1(3)+1
2230 IF I<4 GOTO 2320
2240 I1(4)=I1(3)
2250 I1(4)=I1(4)+1
2260 IF I<5 GOTO 2320
2270 I1(5)=I1(4)
2280 I1(5)=I1(5)+1
2290 IF I<6 GOTO 2320
2300 I1(6)=I1(5)
2310 I1(6)=I1(6)+1
2320 S=I1(1)+I1(2)+I1(3)+I1(4)+I1(5)+I1(6)
2340 IF S>V THEN GOTO 2360
2350 C=C+1:ON I GOTO 2550,2540,2530,2510,2510,2500
2360 IF I>1 GOTO 2380
2370 I1(1)=IN:GOTO 2550
2380 FOR IL=1 TO I-1
2390 S=0:J=I-IL
```

```
2400 IF J-1<=0 THEN GOTO 2440
2410 FOR K=1 TO J-1
2420 S=S+I1(K)
2430 NEXT K
2440 S=S+(IL+1)1(J)+(IL+1)*(IL+2)/2
2450 IF S>V THEN GOTO 2480
2460 I1(I+1-IL)=IN
2470 ON J GOTO 2540,2530,2520,2510,2500
2480 NEXT IL
2490 I1(1)=IN:GOTO 2550
2500 IF I1(6)<IN GOTO 2310
2510 IF I1(5)<IN GOTO 2280
2520 IF I1(4)<IN GOTO 2250
2530 IF I1(3)<IN GOTO 2220
2540 IF I1(2)<IN GOTO 2190
2550 IF I1(1)<IN GOTO 2160
2560 NEXT I
2570 FOX=C/2[IN:QOX=1-FOX
2575 IF FLAG=1 THEN FOX=QOX:QOX=1-FOX
2580 RETURN
2590 REM * * * WILCOXSON SIGNED RANK APPROX DISTIBUTION *
2600 MU=IN*(IN+1)/4
2610 VAR=(2N+1)*(IN+1)N/24
2620 NX=V+0.5
2630 GOSUB 7010
2640 L4=(3N[2+3N-1)/(10N*(IN+1)*(2N+1))
2650 W=(NX-MU)/SQR(VAR):F=(EXP(-0.5*W[2))/SQR(6.28318)
2660 F3=F*(W[3-3*W)
2670 FOX=FOX+L4*F3
2680 F5=-F*(W[5-10*W[3+15*W)
2690 F7=-F*(W[7-21*W[5+105*W[3-105*W)
2700
L6=4*(3N[4+6N[3-3N+1)/(35*(IN*(IN+1)*(2N+1))[2)
2710 FOX=FOX+L6*F5+.5*F7*L4[2
2720 IF FLAG=1 THEN FOX=1-FOX
2730 QOX=1-FOX
2740 RETURN
7000 REM  * * *  NORMAL DISTRIBUTION  * *
7010 SD=SQR(VAR)
7020 XN=(NX-MU)/SD:AX=ABS(XN)
7030 XZ=.3989423*EXP((-AX*AX)/2)
7040 NT=1/(1+.2316419*AX)
7050
QOX=XZ*((.3193815*NT)-(.3565638*NT[2)+(1.781478*NT[3)-(1.821
256*NT[4)+(1.330274*NT[5))
7060 IF XN<0 GOTO 7090
7070 FOX=1-QOX
7080 RETURN
7090 FOX=QOX
7100 QOX=1-FOX
7110 RETURN
```

```
8000 REM * * *  OPTION SELECTION ROUTINE * *
8010 PRINT @JP+5,"NUMBER OF OPTION DESIRED
";LB$;CHR$(176);RB$
8020 ZI=INKEY$:IF ZI="" THEN 8020
8030 PRINT @JP+31,ZI;:IO=VAL(ZI)
8040 IF IO<1 OR IO>KS THEN PRINT @JP,"***  NOT A VALID
OPTION  ***";:FOR I=1 TO 1000:NEXT:PRINT @JP,BL$:GOTO 8010
10000 REM * * * ARRAY SORT * *
10010 L=1 : B(L)=N1+1
10020 J=B(L) 'SET END OF ARRAY SEGMENT
10030 I=M-1  'SET START OF ARRAY SEGMENT
10040 IF J-M < 3 THEN 10230   'HANDLE 1 OR 2 ELEMENTS IN
SPECIAL CASE
10050   M1=INT((I+J)/2)      'SET COMPARE ELEMENTS
10060   REM FIND A LARGE ELEMENT AMONG THE SMALL ONES
10070   I=I+1
10080   IF I=J THEN 10170
10090      IF X(P(I)) <= X(P(M1)) THEN 10070
10100      REM FIND A SMALL ELEMENT AMONG THE LARGE ONES
10110      J=J-1
10120      IF I=J THEN 10170
10130         IF X(P(J)) >= X(P(M1)) THEN 10110
10140         REM EXCHANGE OUT OF PLACE ELEMENTS
10150         H=P(I) : P(I)=P(J) : P(J)=H : GOTO 10070
10160         REM ADJUST COMPARE ELEMENT TO NEW MIDDLE
ELEMENT
10170    IF I > M1 THEN I=I-1
10180    IF J=M1 THEN 10210
10190       H=P(I) : P(I)=P(M1) : P(M1)=H
10200       REM SAVE STARTING POINT FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10210    L=L+1 : B(L)=I
10220    GOTO 10020
10230 IF J-M < 2 THEN 10270
10240    IF X(P(M)) <= X(P(M+1)) THEN 10260
10250       H=P(M) : P(M)=P(M+1) : P(M+1)=H
10260      REM SET BEGIN AND ENDPOINTS FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10270 M=B(L)+1 : L=L-1
10280 IF L>0 THEN GOTO 10020
10290 RETURN
11000 REM * * * READ FILE DIRECTORY * *
11010 CLS:PRINT@20,"WILCOXSON SIGNED RANKS TEST":PRINT
11020 F$="EDITDATA/DIR:"+DN$
11030 OPEN "I",1,F$
11040 INPUT #1,FO
11050 FOR I=1 TO FO
11060     INPUT #1,F$(I)
11070 NEXT I
11080 CLOSE 1
11090 RETURN
```

```
12000 REM * * * PRINT FILE DIRECTORY * *
12010 CLS:PRINT@20,"WILCOXSON SIGNED RANKS TEST":PRINT
12020 FOR I=1 TO FO STEP 4
12030     FOR J=1 TO 4
12040         IF I+J-1>FO THEN 12090
12050         IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
12060         PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
12070     NEXT J
12080 NEXT I
12090 RETURN
14000 REM * * *  AVERAGE RANKS  * *
14010 IF JS=JE THEN S(P(JS),1)=SGN(S(P(JS),1)) * JS :
RETURN
14020 AV=0 : FOR I=JS TO JE : AV=AV+I : NEXT I :
AV=AV/(JE-JS+1)
14030 FOR I=JS TO JE : S(P(I),1)=SGN(S(P(I),1)) * AV : NEXT
I
14040 RETURN
19000 REM * * * ERROR HANDLING ROUTINE  * *
19010 IF ERR=106 AND (SL=2005 OR SL=2410 OR SL=2710 OR 3110
OR SL=3510 OR SL=3910) THEN INPUT "NO FILES ON THIS DRIVE.
PRESS ENTER TO CONTINUE";A$ : SL=0 : RESUME 11080
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0
```

PROGRAM 5, MANN/BAS

```
10 CLEAR 1000
20 ON ERROR GOTO 19000
30 DEFINT P:DEFSTR Z:DEFDBL F,L
40 DIM F$(95),P(3000),X(3000),B(50)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 REM * * *  MANN-WHITNEY TEST  * *
70 CLS:PRINT @20,"MANN-WHITNEY TEST"
75 PRINT:PRINT:PRINT "Disk Drive Containing Data Files
(0-3)?"
80 DN$=INKEY$:IF DN$="" THEN 80 ELSE I=VAL(DN$)
85 IF I<0 OR I>3 THEN GOTO 75
90 SL=100 : GOSUB 11000 : IF SL=0 THEN 70 ELSE GOSUB 12000
: PRINT ""
100 PRINT:INPUT "Enter the Two Data Files (e.g. 3,5)";F1,F2
110 IF F1>F0 OR F2>F0 OR F1<1 OR F2<1 THEN 100
120 OPEN "I",1,F$(F1)+":"+DN$
130 INPUT #1,N
140 FOR I=1 TO N : INPUT #1,X(I) : NEXT I
150 CLOSE 1
160 OPEN "I",1,F$(F2)+":"+DN$
170 INPUT #1,M
180 FOR I=N+1 TO N+M : INPUT #1,X(I) : NEXT I
190 CLOSE 1
200 MT=M
210 N1=N+M:M=1 : FOR I=1 TO N1  : P(I)=I : NEXT I : GOSUB
10000
220 JS=1
230 FOR K=2 TO N1
240 IF X(P(JS))<>X(P(K)) THEN JE=K-1: GOSUB 14000: JS=K
250 NEXT K
260 JE=N1 : GOSUB 14000
270 TS=0
280 FOR I=1 TO N
290  TS=TS+X(I)
300 NEXT I
305 UX=TS-N*(N+1)/2 : UY=NT-UX
310 IF UX < UY THEN UU=UX ELSE UU=UY
320 CLS:PRINT@20,"MANN-WHITNEY TEST":PRINT:PRINT
330 PRINT "HYPOTHESIS          NULL
ALTERNATIVE"
340 PRINT "   (1)         E(X)  =  E(Y)              E(X) =/=
E(Y)"
350 PRINT "   (2)         E(X) <=  E(Y)              E(X)  >
E(Y)"
360 PRINT "   (3)         E(X) =>  E(Y)              E(X)  <
E(Y)"
```

```
370 PRINT:PRINT "Where X and Y represent Data Sets 1 & 2
Respectively"
380 JP=640:KS=3:GOSUB 8000
385 IF IO=1 THEN U=UU ELSE IF IO=2 THEN U=UX ELSE IF IO=3
THEN U=UY
390 CLS:PRINT@20,"MANN:-WHITNEY TEST":PRINT:PRINT
400 PRINT "Test Statistic = ";U
405 PRINT "Sample Sizes are ";N;:PRINT " and ";MT
410 GOSUB 2000
420 ON IO GOSUB 600,800,1000
430 PRINT @960,"Press SPACE BAR when ready to return to
main menu";
440 IF INKEY$= "" GOTO 440 ELSE RUN "MENU/BAS"
600 REM * * * NULL HYPOTHESIS #1 * *
610 PRINT "P-Value = ";:PRINT USING "#.####";2 * FOX
620 PRINT:INPUT"Enter Desired Alpha Level";AL
630 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 620
640 PRINT:PRINT "For Alpha =";AL;", ";
650 IF AL<2 * FOX THEN PRINT "accept"; ELSE PRINT "reject";
660 PRINT " the hypothesis that E(X) = E(Y)"
670 PRINT "-vs- the alternative that E(X) =/= E(Y)."
680 RETURN
800 REM * * * NULL HYPOTHESIS #2 * *
810 PRINT "P-Value = ";:PRINT USING "#.####";FOX
820 PRINT:INPUT"Enter Desired Alpha Level";AL
830 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 620
840 PRINT:PRINT "For Alpha =";AL;", ";
850 IF AL<FOX THEN PRINT "accept"; ELSE PRINT "reject";
860 PRINT " the hypothesis that E(X) <= E(Y)"
870 PRINT "-vs- the alternative that E(X) > E(Y)."
880 RETURN
1000 REM * * * NULL HYPOTHESIS #3 * *
1010 PRINT "P-Value = ";:PRINT USING "#.####";FOX
1020 PRINT:INPUT"Enter Desired Alpha Level";AL
1030 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A
VALUE BETWEEN 0 & 1":GOTO 620
1040 PRINT:PRINT "For Alpha =";AL;", ";
1050 IF AL<FOX THEN PRINT "accept"; ELSE PRINT "reject";
1060 PRINT " the hypothesis that E(X) => E(Y)"
1070 PRINT "-vs- the alternative that E(X) < E(Y)."
1080 RETURN
2000 REM  * * *  MANN-WHITNEY DISTRIBUTION  * *
2010 M=MT
2020 IF M>N THEN M=N:N=MT
2030 FLAG=0
2040 IF U>N/2-.5 THEN U=N-U-1:FLAG=1
2050 IN=N+M:V=TS:I=N
2060 IF U>N OR U<0 OR N<1 OR M<1 THEN PRINT " * * ERROR
*":GOTO 100
```

```
2070 IF V>27 OR M>9 THEN GOSUB 2600 ELSE GOSUB 2090
2080 RETURN
2090 IF INT(V)=V THEN GOSUB 2130 ELSE GOTO 2110
2100 RETURN
2110 V=INT(V):GOSUB 2130:TFOX=FOX:V=V+1:GOSUB
2130:FOX=(FOX+TFOX)/2:QOX=1-FOX
2120 RETURN
2130 I1(2)=0:I1(3)=0:I1(4)=0:I1(5)=0:I1(6)=0
2140 C=0
2150 I1(1)=0
2160 I1(1)=I1(1)+1
2170 IF I<2 GOTO 2310
2180 I1(2)=I1(1)
2190 I1(2)=I1(2)+1
2200 IF I<3 GOTO 2320
2210 I1(3)=I1(2)
2220 I1(3)=I1(3)+1
2230 IF I<4 GOTO 2320
2240 I1(4)=I1(3)
2250 I1(4)=I1(4)+1
2260 IF I<5 GOTO 2320
2270 I1(5)=I1(4)
2280 I1(5)=I1(5)+1
2290 IF I<6 GOTO 2320
2300 I1(6)=I1(5)
2310 I1(6)=I1(6)+1
2320 S=I1(1)+I1(2)+I1(3)+I1(4)+I1(5)+I1(6)
2330 IF S>V THEN GOTO 2350
2340 C=C+1:ON I GOTO 2540,2530,2520,2510,2500,2490
2350 IF I>1 GOTO 2370
2360 I1(1)=IN:GOTO 2540
2370 FOR IL=1 TO I-1
2380 S=0:J=I-IL
2390 IF J-1<=0 THEN GOTO 2430
2400 FOR K=1 TO J-1
2410 S=S+I1(K)
2420 NEXT K
2430 S=S+(IL+1)1(J)+(IL+1)*(IL+2)/2
2440 IF S>V THEN GOTO 2470
2450 I1(I+1-IL)=IN
2460 ON J GOTO 2530,2520,2510,2500,2490
2470 NEXT IL
2480 I1(1)=IN:GOTO 2540
2490 IF I1(6)<IN GOTO 2310
2500 IF I1(5)<IN GOTO 2280
2510 IF I1(4)<IN GOTO 2250
2520 IF I1(3)<IN GOTO 2220
2530 IF I1(2)<IN GOTO 2190
2540 IF I1(1)<IN GOTO 2160
2550 S=1
2560 FOR K=0 TO N-1:S=S*(IN-K)/(N-K):NEXT K
```

```
2570 FOX=C/S:QOX=1-FOX
2580 IF FLAG=1 THEN FOX=QOX:QOX=1-FOX
2590 RETURN
2600 REM * * * MANN-WHITNEY APPROXIMATE DISTRIBUTION * *
2610 MU=N*M/2:VAR=N*M*(N+M+1)/12:NX=U+.5
2620 GOSUB 7010
2630 L4=(M[2+N[2+IN+M*N)/(20*M*N*(IN+1))
2640
L6=(2*(M[4+N[4)+4*M*N*(M[2+N[2)+6*(M[2)*(N[2)+4*(M[3+N[3)+7
*M*N*(M+N)+M[2+N[2+2*M*N-IN)/(210*(M[2)*(N[2)*((IN+1)[2))
2650 W=(NX-MU)/SQR(VAR):F=(EXP(-0.5*W[2))/SQR(6.28318)
2660 F3=F*(W[3-3*W)
2670 F5=-F*(W[5-10*W[3+15*W)
2680 F7=-F*(W[7-21*W[5+105*W[3-105*W)
2690 FOX=FOX+L4 * F3
2700 FOX=FOX+L6 * F5
2710 FOX=FOX+.5 * F7 * L4[2
2720 IF FLAG=1 THEN FOX=1-FOX
2730 QOX=1-FOX
2740 RETURN
7000 REM  * * *  NORMAL DISTRIBUTION  * *
7010 SD=SQR(VAR)
7020 XN=(NX-MU)/SD:AX=ABS(XN)
7030 XZ=.3989423*EXP((-AX*AX)/2)
7040 NT=1/(1+.2316419*AX)
7050
QOX=XZ*((.3193815*NT)-(.3565638*NT[2)+(1.781478*NT[3)-(1.821
256*NT[4)+(1.330274*NT[5))
7060 IF XN<0 GOTO 7090
7070 FOX=1-QOX
7080 RETURN
7090 FOX=QOX
7100 QOX=1-FOX
7110 RETURN
8000 REM * * *  OPTION SELECTION ROUTINE * *
8010 PRINT @JP+5,"NUMBER OF OPTION DESIRED
";LB$;CHR$(176);RB$
8020 ZI=INKEY$:IF ZI="" THEN 8020
8030 PRINT @JP+31,ZI;:IO=VAL(ZI)
8040 IF IO<1 OR IO>KS THEN PRINT @JP,"***  NOT A VALID
OPTION  ***";:FOR I=1 TO 1000:NEXT:PRINT @JP,BL$:GOTO 8010
10000 REM * * * ARRAY SORT * *
10010 L=1 : B(L)=N1+1
10020 J=B(L) 'SET END OF ARRAY SEGMENT
10030 I=M-1  'SET START OF ARRAY SEGMENT
10040 IF J-M < 3 THEN 10230   'HANDLE 1 OR 2 ELEMENTS IN
SPECIAL CASE
10050   M1=INT((I+J)/2)      'SET COMPARE ELEMENTS
10060   REM FIND A LARGE ELEMENT AMONG THE SMALL ONES
10070   I=I+1
10080   IF I=J THEN 10170
```

```
10090       IF X(P(I)) <= X(P(M1)) THEN 10070
10100       REM FIND A SMALL ELEMENT AMONG THE LARGE ONES
10110       J=J-1
10120       IF I=J THEN 10170
10130          IF X(P(J)) >= X(P(M1)) THEN 10110
10140          REM EXCHANGE OUT OF PLACE ELEMENTS
10150          H=P(I) : P(I)=P(J) : P(J)=H : GOTO 10070
10160          REM ADJUST COMPARE ELEMENT TO NEW MIDDLE
ELEMENT
10170     IF I > M1 THEN I=I-1
10180     IF J=M1 THEN 10210
10190        H=P(I) : P(I)=P(M1) : P(M1)=H
10200        REM SAVE STARTING POINT FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10210     L=L+1 : B(L)=I
10220     GOTO 10020
10230 IF J-M < 2 THEN 10270
10240     IF X(P(M)) <= X(P(M+1)) THEN 10260
10250        H=P(M) : P(M)=P(M+1) : P(M+1)=H
10260       REM SET BEGIN AND ENDPOINTS FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10270 M=B(L)+1 : L=L-1
10280 IF L>0 THEN GOTO 10020
10290 RETURN
11000 REM * * * READ FILE DIRECTORY * *
11010 CLS:PRINT@20,"MANN-WHITNEY TEST":PRINT
11020 F$="EDITDATA/DIR:"+DN$
11030 OPEN "I",1,F$
11040 INPUT #1,F0
11050 FOR I=1 TO F0
11060      INPUT #1,F$(I)
11070 NEXT I
11080 CLOSE 1
11090 RETURN
12000 REM * * *  PRINT FILE DIRECTORY * *
12010 CLS:PRINT@20,"MANN-WHITNEY TEST":PRINT
12020 FOR I=1 TO F0 STEP 4
12030      FOR J=1 TO 4
12040          IF I+J-1>F0 THEN 12090
12050          IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
12060          PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
12070      NEXT J
12080 NEXT I
12090 RETURN
14000 REM * * * AVERAGE RANKS * *
14010 IF JS=JE THEN X(P(JS))=JS : RETURN
14020 AV=0:FOR I=JS TO JE: AV=AV+I:NEXT I: AV=AV/(JE-JS+1)
14030 FOR I=JS TO JE:X(P(I))=AV: NEXT I
14040 RETURN
```

```
19000 'ERROR HANDLING
19010 IF ERR=106 AND (SL=2005 OR SL=2410 OR SL=2710 OR 3110
OR SL=3510 OR SL=3910) THEN INPUT "NO FILES ON THIS DRIVE.
PRESS ENTER TO CONTINUE";A$ : SL=0 : RESUME 11080
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0
```

PROGRAM 6. SMIRNOV/BAS

```
10 ON ERROR GOTO 19000
20 DEFSTR Z: DEFINT P
30 DIM F$(95),P(1500),X1(1500),B(50),X2(1500),X(1500)
40 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
50 REM * * *  SMIRNOV TEST  * *
60 CLS:PRINT @20,"SMIRNOV TEST"
70 PRINT:PRINT:PRINT "Disk Drive Containing Data Files
(0-3)?"
80 DN$=INKEY$:IF DN$="" THEN 80 ELSE I=VAL(DN$)
90 IF I<0 OR I>3 THEN GOTO 70
100 SL=110 : GOSUB 11000 : IF SL=0 THEN 60 ELSE GOSUB 13000
: PRINT ""
110 PRINT:INPUT "Enter the Two Data Files (e.g. 3,5)";F1,F2
120 IF F1>F0 OR F2>F0 OR F1<1 OR F2<1 THEN 110
130 OPEN "I",1,F$(F1)+":"+DN$
140 INPUT #1,N
150 FOR I=1 TO N : INPUT #1,X(I) : NEXT I
160 CLOSE 1
170 N1=N:M=1:FOR I=1 TO N:P(I)=I:NEXT I:GOSUB 10000
180 FOR I=1 TO N:X1(I)=X(P(I)):NEXT I
190 OPEN "I",1,F$(F2)+":"+DN$
200 INPUT #1,MM
210 FOR I=1 TO MM : INPUT #1,X(I) : NEXT I
220 CLOSE 1
230 N1=MM:M=1:FOR I=1 TO MM:P(I)=I:NEXT I:GOSUB 10000
240 FOR I=1 TO MM:X2(I)=X(P(I)):NEXT I
242 IF MM=>N THEN GOTO 250
244 FOR I=1 TO N:X(I)=X1(I) : NEXT I
245 FOR I=1 TO MM : X1(I)=X2(I) : NEXT I
246 FOR I=1 TO N : X2(I)=X(I) : NEXT I
248 I=N : N=MM : MM=I
250 I=1:J=1:XCDF=0:YCDF=0
260 TP=0:TM=0
270 IF I>N AND J>MM THEN GOTO 380
280 IF X1(I)=X2(J) THEN I=I+1:J=J+1:GOTO 330
290 IF X1(I) < X2(J) THEN GOTO 300 ELSE GOTO 310
300 IF I <=N THEN I=I+1:GOTO 330:ELSE IF J<=MM THEN
J=J+1:GOTO 330:ELSE GOTO 380
310 IF X1(I) > X2(J) THEN GOTO 320 ELSE GOTO 330
320 IF J <=MM THEN J=J+1 ELSE IF I <=N THEN I=I+1 ELSE GOTO
380
330 YCDF=(I-1)/N : XCDF=(J-1)/MM
340 D=YCDF-XCDF
350 IF D<TM THEN TM=D
360 IF D>TP THEN TP=D
370 GOTO 270
380 CLS:PRINT@20,"SMIRNOV TEST":PRINT:PRINT
```

```
390 PRINT "HYPOTHESIS          NULL
ALTERNATIVE"
400 PRINT "    (1)         F(X)  =  G(X)              F(X) =/=
G(X)"
410 PRINT "    (2)         F(X) <=  G(X)              F(X)  >
G(X)"
420 PRINT "    (3)         F(X) =>  G(X)              F(X)  <
G(X)"
430 PRINT:PRINT "Where F(X) and G(X) represent Data Sets 1
& 2 Respectively"
440 JP=640:KS=3:GOSUB 8000
450 CLS:PRINT@220,"SMIRNOV TEST":PRINT:PRINT
460 ON IO GOTO 470,480,490
470 IF TP>ABS(TM) THEN D=TP ELSE D=ABS(TM):GOTO 500
480 D=TP:GOTO 500
490 D=ABS(TM)
500 PRINT "Test Statistic = ";D
505 PRINT "Sample Sizes are ";N;:PRINT " and ";MM
510 GOSUB 2000
520 ON IO GOSUB 600,800,1000
530 PRINT @960,"Press SPACE BAR when ready to return to
main menu";
540 IF INKEY$= "" GOTO 540 ELSE RUN "MENU/BAS"
600 REM * * * NULL HYPOTHESIS #1 * *
610 PRINT "P-Value = ";:PRINT USING "#.####";2 * QOX
620 PRINT:INPUT"Enter Desired Alpha Level";AL
630 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 620
640 PRINT:PRINT "For Alpha =";AL;", ";
650 IF AL<2*QOX THEN PRINT "accept"; ELSE PRINT "reject";
660 PRINT " the hypothesis that F(X) = G(X)"
670 PRINT "-vs- the alternative that F(X) =/= G(X)."
680 RETURN
800 REM * * * NULL HYPOTHESIS #2 * *
810 PRINT "P-Value = ";:PRINT USING "#.####";QOX
820 PRINT:INPUT"Enter Desired Alpha Level";AL
830 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 620
840 PRINT:PRINT "For Alpha =";AL;", ";
850 IF AL<QOX THEN PRINT "accept"; ELSE PRINT "reject";
860 PRINT " the hypothesis that F(X) <= G(X)"
870 PRINT "-vs- the alternative that F(X) > G(X)."
880 RETURN
1000 REM * * * NULL HYPOTHESIS #3 * *
1010 PRINT "P-Value = ";:PRINT USING "#.####";QOX
1020 PRINT:INPUT"Enter Desired Alpha Level";AL
1030 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A
VLAUE BETWEEN 0 & 1":GOTO 620
1040 PRINT:PRINT "For Alpha =";AL;", ";
1050 IF AL<QOX THEN PRINT "accept"; ELSE PRINT "reject";
1060 PRINT " the hypothesis that F(X) => G(X)"
1070 PRINT "-vs- the alternative that F(X) < G(X)."
```

```
1080 RETURN
2000 REM  * * *  SMIRNOV  DISTRIBUTION  * *
2010 M=MM
2020 IF N<=.1 THEN S=(D-1/(2))*(.12+SQR(N)+.11/SQR(N))
 ELSE IF (M/N)=INT(M/N) THEN S=D * SQR(N/(N+M))
            +2/(3 * SQR(M)) ELSE S=D * SQR(N/(N+M))
            +2/(5 * SQR(M))
2030 QOX=(.135)[(S[2)
2040 IF QOX<0 THEN QOX=0
2050 IF QOX>1 THEN QOX=1
2060 FOX=1-QOX
2070 PX=D
2080 RETURN
8000 REM * * *  OPTION SELECTION ROUTINE * *
8010 PRINT @JP+5,"NUMBER OF OPTION DESIRED
";LB$;CHR$(176);RB$
8020 ZI=INKEY$:IF ZI="" THEN 8020
8030 PRINT @JP+31,ZI;:IO=VAL(ZI)
8040 IF IO<1 OR IO>KS THEN PRINT @JP,"***  NOT A VALID
OPTION  ***";:FOR I=1 TO 1000:NEXT:PRINT @JP,BL$:GOTO 8010
10000 REM * * * ARRAY SORT  * *
10010 L=1 : B(L)=N1+1
10020 J=B(L) 'SET END OF ARRAY SEGMENT
10030 I=M-1  'SET START OF ARRAY SEGMENT
10040 IF J-M < 3 THEN 10230   'HANDLE 1 OR 2 ELEMENTS IN
SPECIAL CASE
10050   M1=INT((I+J)/2)      'SET COMPARE ELEMENTS
10060   REM FIND A LARGE ELEMENT AMONG THE SMALL ONES
10070   I=I+1
10080   IF I=J THEN 10170
10090      IF X(P(I)) <= X(P(M1)) THEN 10070
10100      REM FIND A SMALL ELEMENT AMONG THE LARGE ONES
10110      J=J-1
10120      IF I=J THEN 10170
10130         IF X(P(J)) >= X(P(M1)) THEN 10110
10140         REM EXCHANGE OUT OF PLACE ELEMENTS
10150         H=P(I) : P(I)=P(J) : P(J)=H : GOTO 10070
10160         REM ADJUST COMPARE ELEMENT TO NEW MIDDLE
ELEMENT
10170    IF I > M1 THEN I=I-1
10180    IF J=M1 THEN 10210
10190       H=P(I) : P(I)=P(M1) : P(M1)=H
10200       REM SAVE STARTING POINT FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10210    L=L+1 : B(L)=I
10220    GOTO 10020
10230 IF J-M < 2 THEN 10270
10240    IF X(P(M)) <= X(P(M+1)) THEN 10260
10250       H=P(M) : P(M)=P(M+1) : P(M+1)=H
10260      REM SET BEGIN AND ENDPOINTS FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10270 M=B(L)+1 : L=L-1
```

```
10280 IF L>0 THEN GOTO 10020
10290 RETURN
11000 REM  * * *  READ DATA FILE DIRECTORY  * *
11010 CLS:PRINT@20,"SMIRNOV TEST":PRINT
11020 F$="EDITDATA/DIR:"+DN$
11030 OPEN "I",1,F$
11040 INPUT #1,F0
11050 FOR I=1 TO F0
11060     INPUT #1,F$(I)
11070 NEXT I
11080 CLOSE 1
11090 RETURN
13000 REM  * * *  PRINT DATA FILE DIRECTORY  * *
13010 CLS:PRINT@20,"SMIRNOV TEST":PRINT
13020 FOR I=1 TO F0 STEP 4
13030     FOR J=1 TO 4
13040         IF I+J-1>F0 THEN 13090
13050         IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
13060         PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
13070     NEXT J
13080 NEXT I
13090 RETURN
19000 'ERROR HANDLING
19010 IF ERR=106 AND (SL=2005 OR SL=2410 OR SL=2710 OR 3110
OR SL=3510 OR SL=3910) THEN INPUT "NO FILES ON THIS DRIVE.
PRESS ENTER TO CONTINUE";A$ : SL=0 : RESUME 11080
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0
```

PROGRAM_Z,_NLILL/BAS

```
5 CLEAR 1000
10 ON ERROR GOTO 19000
20 DEFSTR Z: DEFINT P
30 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
40 DIM F$(95),P(1500),X1(1500),B(50),X(1500)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 REM  * * *  LILLIEFOR TEST FOR NORMALITY  * *
70 CLS:PRINT @20,"N O R M A L I T Y     T E S T "
80 PRINT:PRINT:PRINT "Disk Drive Containing Data Files
(0-3)? ";
90 DN$=INKEY$:IF DN$="" THEN 90 ELSE PRINT DN$: I=VAL(DN$)
100 IF I<0 OR I>3 THEN GOTO 80
110 SL=110 : GOSUB 11000 : IF SL=0 THEN 70 ELSE GOSUB 12000
: PRINT ""
120 PRINT:INPUT "Enter the Data File #";F1
130 IF F1>F0 OR F1<1  THEN 120
140 OPEN "I",1,F$(F1)+":"+DN$
150 INPUT #1,N
160 FOR I=1 TO N : INPUT #1,X(I) : NEXT I
170 CLOSE 1
180 CLS:PRINT @458, "* * *  O R D E R I N G     D A T A
* *"
190 N1=N:M=1:FOR I=1 TO N:P(I)=I:NEXT I:GOSUB 10000
200 SM=0:SUM=0:S2=0
210 FOR I=1 TO
N:X1(I)=X(P(I)):SM=SM+X1(I):DV=X1(I)-SUM:SUM=SUM+DV/I:S2=S2+
DV*(X1(I)-SUM):NEXT I
220 CLS:PRINT "SELECT OPTION "
230 PRINT:PRINT:PRINT "(1)   MEAN and VARIANCE are both
unknown"
240 PRINT:PRINT "(2)   MEAN is known, VARIANCE is unknown"
250 PRINT:PRINT "(3)   MEAN and VARIANCE are both known"
260 JP=896:KS=3:GOSUB 8000
290 ON IO GOTO 300,320,340
300 MU=SM/N:VR=S2/(N-1):SD=SQR(VR):XCDF=0:T=0
310 GOTO 370
320 CLS:INPUT "Enter the MEAN";MU
330 VR=0:FOR I=1 TO N:VR=VR+(X1(I)-MU)[2:NEXT
I:SD=SQR(VR/(N-1)):GOTO 370
340 CLS: INPUT "Enter the MEAN";MU
350 INPUT "Enter the VARIANCE";VR:SD=SQR(VR)
360 IF VR<0 THEN PRINT "* * *  ERROR * * *  VARIANCE MUST
BE POSITIVE":GOTO 350
370 FOR I= 1 TO N
380 XN=(X1(I)-MU)/SD:AX=ABS(XN)
390 XZ=.398942EXP((-AX*AX)/2)
400 NT=1/(1+.2316419*AX)
```

```
410
QOX=XZ*((.3193815*NT)-(.3565638*NT[2)+(1.781478*NT[3)-(1.821
256*NT[4)+(1.33274*NT[5))
420 IF XN<O THEN QOX=1-QOX
430 FOX=1-QOX
440 D=ABS(XCDF-FOX)
450 IF D>T THEN T=D
460 XCDF=I/N
470 D=ABS(XCDF-FOX)
480 IF D>T THEN T=D
490 NEXT I
500 CLS:PRINT@20,"N O R M A L I T Y      T E S T "
510 PRINT:PRINT:PRINT "Test Statistic = ";T
520 PRINT "Sample Size = ";N
530 ON IO GOSUB 2000,3000,4000
580 PRINT "P-Value = ";:PRINT USING "#.####";QOX
590 PRINT:INPUT"Enter Desired Alpha Level";AL
600 IF AL<O OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN O & 1":GOTO 590
610 PRINT:PRINT "For Alpha =";AL;", ";
620 IF AL<QOX THEN PRINT "accept"; ELSE PRINT "reject";
630 PRINT " the hypothesis that the sample is from"
640 PRINT "a normal population."
800 PRINT @960,"Press SPACE BAR when ready to return to
main menu";
810 IF INKEY$= "" GOTO 810 ELSE RUN "MENU/BAS"
2000 REM * * * LILLIEFOR'S DISTRIBUTION - MEAN AND VARIANCE
UNKNOWN
2020 DENOM=1
2030 D= T/DENOM
2040 S=D*(-.01+SQR(N)+.85/SQR(N))
2050 TDENOM=DENOM
2060 GOSUB 2500
2070 IF FOX <= .85 THEN DENOM = 1-.24 * FOX:GOTO 2200
2080 IF FOX <= .9 THEN DENOM = 1.357 - .66* FOX : GOTO 2200
2090 IF FOX <= .95 THEN DENOM = 1.339 - .64* FOX : GOTO
2200
2100 IF FOX <= .975 THEN DENOM = 1.795 - 1.12* FOX : GOTO
2200
2110 DENOM = 2.133 - 1.467* FOX
2200 IF ABS(TDENOM - DENOM) > .001 THEN GOTO 2030
2210 RETURN
2500 QOX = .135[(S[2)
2510 IF QOX > 1 THEN QOX = 1
2520 FOX = 1 - QOX
2530 RETURN
2540 IF PR => .70 THEN XQ = 0.743*PR + 0.2799 : GOTO 2590
2550 IF PR => .50 THEN XQ = 0.546*PR + 0.4187 : GOTO 2590
2560 IF PR => .30 THEN XQ = 0.464*PR + 0.4588 : GOTO 2590
2570 IF PR => .20 THEN XQ = 0.485*PR + 0.4525 : GOTO 2590
2580 IF PR => .10 THEN XQ = 0.587*PR + 0.4321 : GOTO 2590
2585 XQ = .846*PR + 0.4062
```

```
2590 XQ=XQ/DENOM
2600 XQ=XQ+.20/N
2610 RETURN
3000 REM * * * LILLIEFOR'S DISTRIBUTION - MEAN KNOWN AND
VARIANCE UNKNOWN : * *
3010 DENOM = 1 : PX = D
3020 DENOM = PX/DENOM
3022 S=D*(.055+SQR(N)+.283/SQR(N))
3026 TDENOM = DENOM
3030 GOSUB 2500
3040 IF FOX <= .9 THEN DENOM = 1.83 -.8* FOX : GOTO 3200
3050 IF FOX <= .95 THEN DENOM = 1.47 - .4* FOX : GOTO 3200
3060 IF FOX <= .975 THEN DENOM = 1.85 - .8* FOX : GOTO 3200
3070 DENOM = 1.07
3200 IF ABS(TDENOM-DENOM) > .001 THEN GOTO 3020
3210 RETURN
4000 REM * * *  KOLMOGOROV DISTRIBUTION * *
4010 S=T*(0.12+SQR(N)+.11/SQR(N))
4020 QOX=.135[(S[2)
4030 FOX=1-QOX
4040 RETURN
8000 REM * * *  OPTION SELECTION ROUTINE * *
8010 PRINT @JP+5,"NUMBER OF OPTION DESIRED
";LB$;CHR$(176);RB$
8020 ZI=INKEY$:IF ZI="" THEN 8020
8030 PRINT @JP+31,ZI;:IO=VAL(ZI)
8040 IF IO<1 OR IO>KS THEN PRINT @JP,"***  NOT A VALID
OPTION  ***";:FOR I=1 TO 1000:NEXT:PRINT @JP,BL$:GOTO 8010
8050 RETURN
10000 REM  * * *  ARRAY SORT  * *
10010 L=1 : B(L)=N1+1
10020 J=B(L) 'SET END OF ARRAY SEGMENT
10030 I=M-1  'SET START OF ARRAY SEGMENT
10040 IF J-M < 3 THEN 10230   'HANDLE 1 OR 2 ELEMENTS IN
SPECIAL CASE
10050   M1=INT((I+J)/2)      'SET COMPARE ELEMENTS
10060   REM FIND A LARGE ELEMENT AMONG THE SMALL ONES
10070   I=I+1
10080   IF I=J THEN 10170
10090      IF X(P(I)) <= X(P(M1)) THEN 10070
10100      REM FIND A SMALL ELEMENT AMONG THE LARGE ONES
10110      J=J-1
10120      IF I=J THEN 10170
10130         IF X(P(J)) >= X(P(M1)) THEN 10110
10140         REM EXCHANGE OUT OF PLACE ELEMENTS
10150         H=P(I) : P(I)=P(J) : P(J)=H : GOTO 10070
10160         REM ADJUST COMPARE ELEMENT TO NEW MIDDLE
ELEMENT
10170    IF I > M1 THEN I=I-1
10180    IF J=M1 THEN 10210
10190       H=P(I) : P(I)=P(M1) : P(M1)=H
10200       REM SAVE STARTING POINT FOR ARRAY SEGMENT OF
```

```
LARGE ELEMENTS
10210     L=L+1 : B(L)=I
10220     GOTO 10020
10230 IF J-M < 2 THEN 10270
10240     IF X(P(M)) <= X(P(M+1)) THEN 10260
10250        H=P(M) : P(M)=P(M+1) : P(M+1)=H
10260       REM SET BEGIN AND ENDPOINTS FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10270 M=B(L)+1 : L=L-1
10280 IF L>0 THEN GOTO 10020
10290 RETURN
11000 REM  * * * READ DATA FILE DIRECTORY  * *
11010 CLS:PRINT@20,"N O R M A L I T Y     T E S T":PRINT
11020 F$="EDITDATA/DIR:"+DN$
11030 OPEN "I",1,F$
11040 INPUT #1,FO
11050 FOR I=1 TO FO
11060     INPUT #1,F$(I)
11070 NEXT I
11080 CLOSE 1
11090 RETURN
12000 REM  * * *  LIST DATA FILE DIRECTORY  * *
12010 CLS:PRINT@20,"N O R M A L I T Y     T E S T":PRINT
12020 FOR I=1 TO FO STEP 4
12030     FOR J=1 TO 4
12040         IF I+J-1>FO THEN 12090
12050         IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
12060         PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
12070     NEXT J
12080 NEXT I
12090 RETURN
19000 'ERROR HANDLING
19010 IF ERR=106 AND (SL=2005 OR SL=2410 OR SL=2710 OR 3110
OR SL=3510 OR SL=3910) THEN INPUT "NO FILES ON THIS DRIVE.
PRESS ENTER TO CONTINUE";A$ : SL=0 : RESUME 11080
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0
```

PROGRAM 8. ELILL/BAS

```
5 CLEAR 1000
10 ON ERROR GOTO 19000
20 DEFSTR Z: DEFINT P
30 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
40 DIM F$(95),P(1500),X1(1500),B(50),X(1500)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 REM  * * *  EXPONENTIAL DISTRIBUTION TEST  * *
70 CLS:PRINT @20,"E X P O N E N T I A L    T E S T "
80 PRINT:PRINT:PRINT "Disk Drive Containing Data Files
(0-3)? ";
90 DN$=INKEY$:IF DN$="" THEN 90 ELSE PRINT DN$: I=VAL(DN$)
100 IF I<0 OR I>3 THEN GOTO 80
110 SL=110 : GOSUB 11000 : IF SL=0 THEN 70 ELSE GOSUB 12000
: PRINT ""
120 PRINT:INPUT "Enter the Data File #";F1
130 IF F1>F0 OR F1<1  THEN 120
140 OPEN "I",1,F$(F1)+":"+DN$
150 INPUT #1,N
160 FOR I=1 TO N : INPUT #1,X(I) : NEXT I
170 CLOSE 1
180 CLS:PRINT @458, "* * *  O R D E R I N G     D A T A
* *"
190 N1=N:M=1:FOR I=1 TO N:P(I)=I:NEXT I:GOSUB 10000
200 SM=0
210 FOR I=1 TO N:X1(I)=X(P(I)):SM=SM+X1(I):NEXT I:LA=N/SM
220 CLS:PRINT "SELECT OPTION "
230 PRINT:PRINT:PRINT "(1)   LAMBDA is unknown"
240 PRINT:PRINT "(2)   LAMBDA is known"
250 JP=896:KS=2:GOSUB 8000
260 ON IO GOTO 290,270
270 CLS:INPUT "Enter LAMBDA";LA
280 IF LA <= 0 THEN PRINT "* * *  ERROR  * * *  LAMBDA MUST
BE POSITIVE":GOTO 270
290 T=0:XCDF=0
300 FOR I=1 TO N
310 FOX = 1-EXP(-LA1(I))
320 D=ABS(XCDF-FOX)
330 IF D > T THEN T=D
340 XCDF = I/N
350 D=ABS(XCDF-FOX)
360 IF D > T THEN T=D
370 NEXT I
380 CLS:PRINT@20,"E X P O N E N T I A L    T E S T "
390 PRINT:PRINT:PRINT "Test Statistic = ";T
400 PRINT "Sample Size = ";N
410 ON IO GOSUB 2000,3000
420 PRINT "P-Value = ";:PRINT USING "#.####";@OX
```

```
430 PRINT:INPUT"Enter Desired Alpha Level";AL
440 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 430
450 PRINT:PRINT "For Alpha =";AL;", ";
460 IF AL<QOX THEN PRINT "accept"; ELSE PRINT "reject";
470 PRINT " the hypothesis that the sample is from"
480 PRINT "a normal population."
490 PRINT @960,"Press SPACE BAR when ready to return to
main menu";
500 IF INKEY$= "" GOTO 500 ELSE RUN "MENU/BAS"
2000 REM * * * LILLIEFOR'S DISTRIBUTION - MEAN AND VARIANCE
UNKNOWN
2020 DENOM=SQR(N)+.26+.5/SQR(N)
2040 IF T > 1.3442/DENOM + .2/N THEN FOX = 1 : GOTO 2220
2060 FOX=.5:GOSUB 2280
2080 IF T>XQ THEN F1=FOX:F2=1:GOTO 2120
2100 F1=0 : F2=FOX
2120 FOX = (F1+F2)/2
2140 GOSUB 2280
2160 IF ABS(T-XQ)<.0001 THEN GOTO 2220
2180 IF T>XQ THEN F1=FOX ELSE F2=FOX
2200 GOTO 2120
2220 QOX=1-FOX
2230 RETURN
2280 IF FOX => .95 THEN XQ = 5.125* FOX - 3.7808 : GOTO
2460
2300 IF FOX => .90 THEN XQ = 2.026* FOX - 0.8367 : GOTO
2460
2320 IF FOX => .80 THEN XQ = 1.124* FOX - 0.0249 : GOTO
2460
2340 IF FOX => .70 THEN XQ = 0.743* FOX + 0.2799 : GOTO
2460
2360 IF FOX => .50 THEN XQ = 0.546* FOX + 0.4187 : GOTO
2460
2380 IF FOX => .30 THEN XQ = 0.464* FOX + 0.4588 : GOTO
2460
2400 IF FOX => .20 THEN XQ = 0.485* FOX + 0.4525 : GOTO
2460
2420 IF FOX => .10 THEN XQ = 0.587* FOX + 0.4321 : GOTO
2460
2440 XQ = .846* FOX + 0.4062
2460 XQ=XQ/DENOM + .2/N
2470 RETURN
3000 REM * * * LILLIEFOR'S DISTRIBUTION - LAMBA KNOWN : *

3010 S=T*(0.12+SQR(N)+.11/SQR(N))
3020 QOX=.135[(S[2)
3030 IF QOX>1 THEN QOX=1
3040 FOX=1-QOX
3050 PX=T
3060 RETURN
8000 REM * * *  OPTION SELECTION ROUTINE * *
```

```
8010 PRINT @JP+5,"NUMBER OF OPTION DESIRED
";LB$;CHR$(176);RB$
8020 ZI=INKEY$:IF ZI="" THEN 8020
8030 PRINT @JP+31,ZI;:IO=VAL(ZI)
8040 IF IO<1 OR IO>KS THEN PRINT @JP,"***  NOT A VALID
OPTION  ***";:FOR I=1 TO 1000:NEXT:PRINT @JP,BL$:GOTO 8010
8050 RETURN
10000 REM  * * *  ARRAY SORT  * *
10010 L=1 : B(L)=N1+1
10020 J=B(L) 'SET END OF ARRAY SEGMENT
10030 I=M-1  'SET START OF ARRAY SEGMENT
10040 IF J-M < 3 THEN 10230   'HANDLE 1 OR 2 ELEMENTS IN
SPECIAL CASE
10050   M1=INT((I+J)/2)      'SET COMPARE ELEMENTS
10060   REM FIND A LARGE ELEMENT AMONG THE SMALL ONES
10070   I=I+1
10080   IF I=J THEN 10170
10090      IF X(P(I)) <= X(P(M1)) THEN 10070
10100      REM FIND A SMALL ELEMENT AMONG THE LARGE ONES
10110      J=J-1
10120      IF I=J THEN 10170
10130         IF X(P(J)) >= X(P(M1)) THEN 10110
10140         REM EXCHANGE OUT OF PLACE ELEMENTS
10150         H=P(I) : P(I)=P(J) : P(J)=H : GOTO 10070
10160         REM ADJUST COMPARE ELEMENT TO NEW MIDDLE
ELEMENT
10170    IF I > M1 THEN I=I-1
10180    IF J=M1 THEN 10210
10190       H=P(I) : P(I)=P(M1) : P(M1)=H
10200       REM SAVE STARTING POINT FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10210    L=L+1 : B(L)=I
10220    GOTO 10020
10230 IF J-M < 2 THEN 10270
10240    IF X(P(M)) <= X(P(M+1)) THEN 10260
10250       H=P(M) : P(M)=P(M+1) : P(M+1)=H
10260      REM SET BEGIN AND ENDPOINTS FOR ARRAY SEGMENT OF
LARGE ELEMENTS
10270 M=B(L)+1 : L=L-1
10280 IF L>0 THEN GOTO 10020
10290 RETURN
11000 REM  * * *  READ DATA FILE DIRECTORY  * *
11010 CLS:PRINT@20,"E X P O N E N T I A L    T E S T":PRINT
11020 F$="EDITDATA/DIR:"+DN$
11030 OPEN "I",1,F$
11040 INPUT #1,FO
11050 FOR I=1 TO FO
11060     INPUT #1,F$(I)
11070 NEXT I
11080 CLOSE 1
11090 RETURN
12000 REM  * * *  LIST DATA FILE DIRECTORY  * *
```

```
12010 CLS:PRINT@20,"E X P O N E N T I A L    T E S T":PRINT
12020 FOR I=1 TO F0 STEP 4
12030     FOR J=1 TO 4
12040          IF I+J-1>F0 THEN 12090
12050          IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
12060          PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
12070     NEXT J
12080 NEXT I
12090 RETURN
19000 'ERROR HANDLING
19010 IF ERR=106 AND (SL=2005 OR SL=2410 OR SL=2710 OR 3110
OR SL=3510 OR SL=3910) THEN INPUT "NO FILES ON THIS DRIVE.
PRESS ENTER TO CONTINUE";A$ : SL=0 : RESUME 11080
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0
```

PROGRAM 9. CHISQU/BAS

```
5 CLEAR 1000
10 ON ERROR GOTO 19000
20 DEFSTR Z: DEFINT P
30 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
40 DIM F$(95)
50 LB$=CHR$(123):RB$=CHR$(125):MB$=CHR$(176):BL$=CHR$(252)
60 REM * * *  CHI-SQUARE TEST * *
70 CLS:PRINT @20,"C H I - S Q U A R E     T E S T"
80 PRINT:PRINT:PRINT "Disk Drive Containing Data Files
(0-3)? ";
90 DN$=INKEY$:IF DN$="" THEN 90 ELSE PRINT DN$: I=VAL(DN$)
100 IF I<0 OR I>3 THEN GOTO 80
110 SL=110 : GOSUB 11000 : IF SL=0 THEN 70 ELSE GOSUB 12000
: PRINT ""
120 PRINT:INPUT "Enter the Data File #";F1
130 IF F1>F0 OR F1<1  THEN 120
140 OPEN "I",1,F$(F1)+":"+DN$
150 INPUT #1,RT:INPUT #1,CT:INPUT #1,BT
160 DIM X(RT,CT),R(RT),C(CT)
170 FOR R1=1 TO RT
180 FOR C1=1 TO CT
190 FOR B1=1 TO BT
200 INPUT #1,X(R1,C1)
210 NEXT B1
220 NEXT C1
230 NEXT R1
240 N=0:DF=(RT-1)*(CT-1)
250 FOR I=1 TO RT
260 R(I)=0
270 FOR J=1 TO CT
280 R(I)=R(I)+X(I,J)
290 NEXT J
300 N=N+R(I)
310 NEXT I
320 FOR J=1 TO CT
330 C(J)=0
340 FOR I=1 TO RT
350 C(J)=C(J)+X(I,J)
360 NEXT I
370 NEXT J
380 T=0:CZ=0:CF=0
390 FOR I=1 TO RT
400 FOR J=1 TO CT
410 E=R(I)* C(J)/N
420 T=T+((X(I,J)-E)[2)/E
430 IF E < 5 THEN CF=CF+1
440 IF E < 1 THEN CZ=CZ+1
```

```
450 NEXT J
460 NEXT I
465 IF RT=2 AND CT=2 THEN T=((ABS(X(1,1)* X(2,2)-X(1,2)
X(2,1))-0.5*N)[2)* N/(R(1)* R(2)* C(1)* C(2))
470 CLS:PRINT @20,"C H I - S Q U A R E     T E S T"
480 PRINT:PRINT:PRINT "Test Statistic = ";T
490 PRINT "Degrees of Freedom =";DF
500 PRINT "Cells with expected frequency < 5 is ";100
CF/(RT* CT);:PRINT "%"
510 PRINT "Number of cells with expected frequency < 1 is
";CZ
520 GOSUB 2000
530 PRINT "P-Value = ";:PRINT USING "#.####";QOX
540 PRINT:INPUT"Enter Desired Alpha Level";AL
550 IF AL<0 OR AL>1 THEN PRINT "*** ERROR *** ENTER A VLAUE
BETWEEN 0 & 1":GOTO 540
560 PRINT:PRINT "For Alpha =";AL;", ";
570 IF AL<QOX THEN PRINT "accept"; ELSE PRINT "reject";
580 PRINT " the hypothesis that the column cells"
590 PRINT "have equal probabilities."
600 PRINT @960,"Press SPACE BAR when ready to return to
main menu";
610 IF INKEY$= "" GOTO 610 ELSE RUN "MENU/BAS"
2000 REM * * *  CHI-SQUARE DISTRIBUTION * *
2020 CX=T
2040 IF DF>30 GOTO 2460
2060 DP=1
2080 FOR I=INT(DF) TO 2 STEP -2
2100 DP=DP
2120 NEXT I
2140 NP=CX[(INT((DF+1)/2))*EXP(-CX/2)/DP
2160 IF INT(DF/2)=DF/2 GOTO 2220
2180 L1=SQR(2/CX/3.141593)
2200 GOTO 2240
2220 L1=1
2240 LC=1
2260 MC=1
2280 D=DF
2300 D=D+2
2320 MC=MC*CX/D
2340 IF MC<.000001 GOTO 2400
2360 LC=LC+MC
2380 GOTO 2300
2400 FOX=L1*NP* LC
2420 QOX=1-FOX
2440 RETURN
2460 NX=((CX/DF)[(1/3) - (1-(2/(9* DF))))/SQR(2/(9* DF))
2480 MU=0 : VAR=1 : I2=1
3000 REM   * * *   N O R M A L   D I S T R I B U T I O N
* *
3010 SD=SQR(VAR)
3020 XN=(NX-MU)/SD:AX=ABS(XN)
```

```
3030 XZ=.3989423* EXP((-AX*AX)/2)
3040 NT=1/(1+.2316419* AX)
3050
QOX=XZ*((.3193815*NT)-(.3565638*NT[2)+(1.781478*NT[3)-(1.821
256*NT[4)+(1.330274*NT[5))
3060 IF XN<0 GOTO 3090
3070 FOX=1-QOX
3080 RETURN
3090 FOX=QOX
3100 QOX=1-FOX
3110 RETURN
11000 REM  * * *  READ DATA FILE DIRECTORY  * *
11010 CLS:PRINT @20,"C H I - S Q U A R E     T E S T":PRINT
11020 F$="CTABLE/DIR:"+DN$
11030 OPEN "I",1,F$
11040 INPUT #1,FO
11050 FOR I=1 TO FO
11060     INPUT #1,F$(I)
11070 NEXT I
11080 CLOSE 1
11090 RETURN
12000 REM   * * *  PRINT OUT DIRECTORY OF DATA FILES  * *
12010 CLS:PRINT @20,"C H I - S Q U A R E     T E S T":PRINT
12020 FOR I=1 TO FO STEP 4
12030     FOR J=1 TO 4
12040         IF I+J-1>FO THEN 12090
12050         IF J=1 AND LEN(F$(I+J-1))<12 AND I>1 THEN
PRINT ""
12060         PRINT TAB(16*(J-1));"(";STR$(I+J-1);")
";F$(I+J-1);
12070     NEXT J
12080 NEXT I
12090 RETURN
19000 'ERROR HANDLING
19010 IF ERR=106 AND (SL=2005 OR SL=2410 OR SL=2710 OR 3110
OR SL=3510 OR SL=3910) THEN INPUT "NO FILES ON THIS DRIVE.
PRESS ENTER TO CONTINUE";A$ : SL=0 : RESUME 11080
19100 PRINT "ERR=";ERR;"ERL=";ERL : ON ERROR GOTO 0
```

PROGRAM 10. NPSTAT/BAS

```
10 CLEAR 350
20 DEFINT I-K : DEFSTR Z
30 DEFDBL F,L
40 LB$=CHR$(123) : RB$=CHR$(125) : MB$=CHR$(176) :
BL$=CHR$(252)
50 CLS : PRINT TAB(20);"***  MENU  ***"
60 PRINT"(1) Probability Distributions
70 PRINT"(2) Inverse Probability Distributions
80 PRINT "(3) Return to Program Selection Menu"
90 KP=704 : KS=3  : GOSUB 130
100 ON IO GOSUB 500,1000,120
110 GOTO 50
120 RUN "MENU/BAS"
130 REM
140 PRINT @KP+5,"OPTION DESIRED ";LB$;MB$;RB$;
150 ZI=INKEY$ : IF ZI="" THEN 150ELSE IO=VAL(ZI)
160 PRINT @KP+21, ZI;: FOR I=1 TO 100 : NEXT
170 IF IO<1 OR IO>KS THEN PRINT @KP, BL$;: PRINT @KP,"**
NOT A VALID OPTION   ***";: FOR I=1 TO 1000 : NEXT : PRINT
@KP, BL$; : GOTO 140
180 RETURN
190 PRINT @KP+5,LB$;" ";L$;" or ";R$;" ";RB$;"
";LB$;MB$;RB$;
200 ZQ=INKEY$ : IF ZQ="" THEN 200ELSE PRINT @KP+18,ZQ;: FOR
I=1 TO 100 : NEXT
210 IF ZQ<>L$ AND ZQ<>CHR$(ASC(L$)-32) AND ZQ<>R$ AND
ZQ<>CHR$(ASC(R$)-32) THEN 190
220 RETURN
500 REM
510 CLS : PRINT TAB(18)"PROBABILITY DISTRIBUTIONS
520 PRINT:PRINT: I2=0
530 PRINT"MENU:"
540 PRINT"(1)  Normal Distribution
550 PRINT"(2)  Wilcoxson Signed-Ranks Distribution"
560 PRINT "(3)  Mann-Whitney Distribution"
570 PRINT "(4)  Kologmorov Distribution"
580 PRINT "(5)  Smirnov Distribution"
590 PRINT "(6)  Lillifor's Normality Test Distribution"
600 PRINT "(7)  Lilliefor's Exponentiality Test
Distribution"
610 PRINT "(8)  Chi-Square Distribution"
620 KP=896 : KS=8 : GOSUB 130
630 ON IO GOSUB 2000,3000,5000,6000,7000,8000,9000,10000
640 IF FLAG=1 THEN FLAG=0:FOX=1-FOX:QOX=1-QOX
650 PRINT : PRINT : PRINT "Prob. ( X <=";PX;") = "; : PRINT
USING"#.####";FOX
660 PRINT "Prob. ( X > ";PX;") = "; : PRINT
```

```
USING"#.####";QOX
670 PRINT @960,"  Run Again or Quit?";: KP=978 : L$="r" :
R$="q" : GOSUB 190
680 CLS : I2=0 : IF ZQ="r" OR ZQ="R" GOTO 630ELSE RETURN
690 GOTO 50
1000 CLS : PRINT TAB(15)"INVERSE PROBABILITY DISTRIBUTIONS
1010 PRINT @192,"MENU:":PRINT"(1)  Inverse Normal
1020 PRINT"(2)  Inverse Wilcoxson-Signed Ranks"
1030 PRINT "(3)  Inverse Mann-Whitney Distribution"
1040 PRINT "(4)  Inverse Kolmogorov Distribution"
1050 PRINT "(5)  Inverse Smirnov Distribution"
1060 PRINT "(6)  Inverse Lilliefor Normality Test
Distribution"
1070 PRINT "(7)  Inverse Lilliefor Exponential Test
Distribution"
1080 PRINT "(8)  Inverse Chi-Square Distribution"
1090 KP=896:KS=8:GOSUB 130
1100 I2=0
1110 ON IO GOSUB
13000,14000,15000,16000,17000,18000,19000,19220
1120 PRINT : PRINT "For  F(x) = Prob. ( X <= x ) =";PR
1130 IF XQ=9E36 THEN PRINT "x =  infinity" ELSE IF XQ=9E-36
PRINT "x =  - infinity"
1140 IF XQ=9E36 OR XQ=9E-36 THEN GOTO 1160
1150 PRINT "x = "; : IF ABS(XQ)>1000 THEN PRINT INT(XQ)
ELSE IF ABS(XQ)>100 PRINT USING"####.#";XQ ELSE IF
ABS(XQ)>10 PRINT USING"###.##";XQ ELSE PRINT
USING"##.####";XQ
1160 PRINT @960,"  Run Again or Quit";: L$="r" : R$="q" :
KP=976 : GOSUB 190
1170 IF ZQ="R" OR ZQ="r" GOTO 1100ELSE RETURN
2000 REM  * * *  NORMAL DISTRIBUTION  * *
2010 IF I2=1 GOTO 2070
2020 CLS:PRINT @20,"NORMAL DISTRIBUTION"
2030 PRINT:PRINT:INPUT"Enter Mean";MU
2040 INPUT"Enter Variance";VAR
2050 IF VAR<=0 PRINT"** ERROR **   ";VAR"IS NOT A VALID
VARIANCE":GOTO 2040
2060 INPUT"Enter Normal Statistic  x ";NX : IF I2=0 THEN
PX=NX
2070 SD=SQR(VAR)
2080 XN=(NX-MU)/SD:AX=ABS(XN)
2090 XZ=.3989423XP((-AX*AX)/2)
2100 NT=1/(1+.2316419*AX)
2110
QOX=XZ*((.3193815*NT)-(.3565638*NT[2)+(1.781478*NT[3)-(1.821
256*NT[4)+(1.330274*NT[5))
2120 IF XN<0 GOTO 2150
2130 FOX=1-QOX
2140 RETURN
2150 FOX=QOX
2160 QOX=1-FOX
```

```
2170 RETURN
3000 REM * * *  WILCOXSON SIGNED-RANKS DISTRIBUTION  * *
3010 CLS:PRINT@10,"WILCOXSON SIGNED-RANKS DISTRIBUTION"
3020 PRINT:PRINT
3030 INPUT "ENTER Wilcoxson Statistic";V:PX=V
3040 IF V < 0 THEN PRINT "* * * ERROR * * *  STATISTIC MUST
BE POSITIVE":GOTO 3030
3050 INPUT "ENTER Sample Size";IN
3060 IF IN < 1 THEN PRINT "* * * ERROR * * *  SAMPLE SIZE
MUST BE > 1":GOTO 3050
3070 IF V>IN*(IN+1)/2 THEN PRINT " * * ERROR * *  ":GOTO
3030
3080 IF V>IN*(IN+1)/4 THEN V=IN*(IN+1)/2-V-1:FLAG=1
3090 IF V>27 OR (IN>9 AND V>IN) THEN GOSUB 3620 ELSE GOSUB
3110
3100 RETURN
3110 IF INT(V)=V GOSUB 3150 ELSE GOTO 3130
3120 RETURN
3130 V=INT(V):GOSUB 3150:TFOX=FOX:V=V+1:GOSUB
3150:FOX=(FOX+TFOX)/2:QOX=1-FOX
3140 RETURN
3150 I1(2)=0:I1(3)=0:I1(4)=0:I1(5)=0:I1(6)=0
3160 C=1
3170 N=INT(.5*(-1+SQR(1+8*V)))
3180 FOR I=1 TO N
3190 I1(1)=0
3200 I1(1)=I1(1)+1
3210 IF I<2 GOTO 3360
3220 I1(2)=I1(1)
3230 I1(2)=I1(2)+1
3240 IF I<3 GOTO 3360
3250 I1(3)=I1(2)
3260 I1(3)=I1(3)+1
3270 IF I<4 GOTO 3360
3280 I1(4)=I1(3)
3290 I1(4)=I1(4)+1
3300 IF I<5 GOTO 3360
3310 I1(5)=I1(4)
3320 I1(5)=I1(5)+1
3330 IF I<6 GOTO 3360
3340 I1(6)=I1(5)
3350 I1(6)=I1(6)+1
3360 S=I1(1)+I1(2)+I1(3)+I1(4)+I1(5)+I1(6)
3370 IF S>V THEN GOTO 3390
3380 C=C+1:ON I GOTO 3580,3570,3560,3540,3540,3530
3390 IF I>1 GOTO 3410
3400 I1(1)=IN:GOTO 3580
3410 FOR IL=1 TO I-1
3420 S=0:J=I-IL
3430 IF J-1<=0 THEN GOTO 3470
3440 FOR K=1 TO J-1
3450 S=S+I1(K)
```

```
3460 NEXT K
3470 S=S+(IL+1)1(J)+(IL+1)*(IL+2)/2
3480 IF S>V THEN GOTO 3510
3490 I1(I+1-IL)=IN
3500 ON J GOTO 3570,3560,3550,3540,3530
3510 NEXT IL
3520 I1(1)=IN:GOTO 3580
3530 IF I1(6)<IN GOTO 3350
3540 IF I1(5)<IN GOTO 3320
3550 IF I1(4)<IN GOTO 3290
3560 IF I1(3)<IN GOTO 3260
3570 IF I1(2)<IN GOTO 3230
3580 IF I1(1)<IN GOTO 3200
3590 NEXT I
3600 FOX=C/2[IN:QOX=1-FOX
3610 RETURN
3620 REM * * * WILCOXSON SIGNED RANK APPROX DISTIBUTION *
3630 MU=IN*(IN+1)/4
3640 VAR=(2N+1)*(IN+1)N/24
3650 NX=V+0.5
3660 GOSUB 2070
3670 L4=(3N[2+3N-1)/(10N*(IN+1)*(2N+1))
3680 W=(NX-MU)/SQR(VAR):F=(EXP(-0.5*W[2))/SQR(6.28318)
3690 F3=F*(W[3-3*W)
3700 FOX=FOX+L4* F3
3710 F5=-F*(W[5-10*W[3+15*W)
3720 F7=-F*(W[7-21*W[5+105*W[3-105*W)
3730
L6=4*(3N[4+6N[3-3N+1)/(35*(IN*(IN+1)*(2N+1))[2)
3740 FOX=FOX+L6* F5+.5*F7* L4[2
3750 QOX=1-FOX
3760 RETURN
5000 REM  * * *  MANN-WHITNEY DISTRIBUTION  * *
5010 CLS:PRINT@10, "MANN-WHITNEY DISTRIBUTION"
5020 PRINT:PRINT
5030 INPUT "ENTER Mann-Whitney Statistic";U:PX=U
5040 IF U < 0 THEN PRINT "* * *  ERROR  * * *  STATISTIC
MUST BE POSITIVE ":GOTO 5030
5050 INPUT "ENTER Smaller Sample Size";N
5060 IF N < 1 THEN PRINT "* * * ERROR * * * SAMPLE SIZE
MUST BE > 1":GOTO 5050
5070 INPUT "ENTER Larger Sample Size";M
5080 IF M < N THEN PRINT " * * *  ERROR * * * YOU ENTERED
THE SMALLER SAMPLE SIZE LAST ":GOTO 5050
5090 FLAG=0
5100 IF U>N*M/2-.5 THEN U=N*M-U-1:FLAG=1
5110 IN=N+M:V=U+N*(N+1)/2:I=N
5120 IF U>N*M  THEN PRINT " * * ERROR * *  THE STATISTIC
MUST BE > THE PRODUCT OF THE SAMPLE SIZES":GOTO 5030
5130 IF V>27 OR M>9 THEN GOSUB 5650 ELSE GOSUB 5150
5140 RETURN
5150 IF INT(V)=V THEN GOSUB 5190 ELSE GOTO 5170
```

```
5160 RETURN
5170 V=INT(V):GOSUB 5190:TFOX=FOX:V=V+1:GOSUB
5190:FOX=(FOX+TFOX)/2:QOX=1-FOX
5180 RETURN
5190 I1(2)=0:I1(3)=0:I1(4)=0:I1(5)=0:I1(6)=0
5200 C=0
5210 I1(1)=0
5220 I1(1)=I1(1)+1
5230 IF I<2 GOTO 5370
5240 I1(2)=I1(1)
5250 I1(2)=I1(2)+1
5260 IF I<3 GOTO 5380
5270 I1(3)=I1(2)
5280 I1(3)=I1(3)+1
5290 IF I<4 GOTO 5380
5300 I1(4)=I1(3)
5310 I1(4)=I1(4)+1
5320 IF I<5 GOTO 5380
5330 I1(5)=I1(4)
5340 I1(5)=I1(5)+1
5350 IF I<6 GOTO 5380
5360 I1(6)=I1(5)
5370 I1(6)=I1(6)+1
5380 S=I1(1)+I1(2)+I1(3)+I1(4)+I1(5)+I1(6)
5390 IF S>V THEN GOTO 5410
5400 C=C+1:ON I GOTO 5600,5590,5580,5570,5550,5550
5410 IF I>1 GOTO 5430
5420 I1(1)=IN:GOTO 5600
5430 FOR IL=1 TO I-1
5440 S=0:J=I-IL
5450 IF J-1<=0 THEN GOTO 5490
5460 FOR K=1 TO J-1
5470 S=S+I1(K)
5480 NEXT K
5490 S=S+(IL+1)1(J)+(IL+1)*(IL+2)/2
5500 IF S>V THEN GOTO 5530
5510 I1(I+1-IL)=IN
5520 ON J GOTO 5590,5580,5570,5560,5550
5530 NEXT IL
5540 I1(1)=IN:GOTO 5600
5550 IF I1(6)<IN GOTO 5370
5560 IF I1(5)<IN GOTO 5340
5570 IF I1(4)<IN GOTO 5310
5580 IF I1(3)<IN GOTO 5280
5590 IF I1(2)<IN GOTO 5250
5600 IF I1(1)<IN GOTO 5220
5610 S=1
5620 FOR K=0 TO N-1:S=S*(IN-K)/(N-K):NEXT K
5630 FOX=C/S:QOX=1-FOX
5640 RETURN
5650 REM * * * MANN-WHITNEY APPROXIMATE DISTRIBUTION * *
5660 MU=N/2:VAR=N*(N+M+1)/12:NX=U+.5
```

```
5670 GOSUB 2070
5680 L4=(M[2+N[2+IN+M*N)/(20*N*(IN+1))
5690
L6=(2*(M[4+N[4)+4*N*(M[2+N[2)+6*(M[2)*(N[2)+4*(M[3+N[3)+7
*N*(M+N)+M[2+N[2+2*N-IN)/(210*(M[2)*(N[2)*((IN+1)[2))
5700 W=(NX-MU)/SQR(VAR):F=(EXP(-0.5[2))/SQR(6.28318)
5710 F3=F*(W[3-3)
5720 F5=-F*(W[5-10[3+15)
5730 F7=-F*(W[7-21[5+105[3-105)
5740 FOX=FOX+L4* F3
5750 FOX=FOX+L6* F5
5760 FOX=FOX+.5* F7* L4[2
5770 RETURN
6000 REM  * * *  KOLMOGROV DISTRIBUTION  * *
6010 CLS:PRINT@10, "KOLMOGOROV DISTRIBUTION"
6020 PRINT:PRINT
6030 INPUT "ENTER SKolmogorov Statistic";D
6040 IF D<0 OR D>1 THEN PRINT "* * ERROR * *":GOTO 6030
6050 INPUT "ENTER Sample Size";N
6060 IF N<1 THEN PRINT "* * ERROR * *":GOTO 6050
6070 S=D*(0.12+SQR(N)+0.11/SQR(N))
6080 GOSUB 6110
6090 PX=D
6100 RETURN
6110 QOX=(.135)[(S[2)
6120 IF QOX<0 THEN QOX=0
6130 IF QOX>1 THEN QOX=1
6140 FOX=1-QOX
6150 RETURN
7000 REM  * * *  SMIRNOV DISTRIBUTION  * *
7010 CLS:PRINT@10, "SMIRNOV DISTRIBUTION"
7020 PRINT:PRINT
7030 INPUT "ENTER Smirnov Statistic";D
7040 IF D<0 OR D>1 THEN PRINT "* * ERROR * *":GOTO 7030
7050 INPUT "ENTER Smaller Sample Size";N
7060 IF N<1 THEN PRINT "* * ERROR * *  SAMPLE SIZE MUST
BE > 1":GOTO 7050
7070 INPUT "ENTER Larger Sample Size";M
7080 IF M<N THEN PRINT "* * ERROR * *  YOU ENTERED THE
SMALLER SAMPLE SIZE LAST":GOTO 7050
7090 IF N<=.1 THEN S=(D-1/(2))*(.12+SQR(N)+.11/SQR(M))
          ELSE IF (M/N)=INT(M/N) THEN
               S=D* SQR(N/(N+M))+2/(3* SQR(M))
          ELSE S=D* SQR(N/(N+M))+2/(5* SQR(M))
7100 GOSUB 6110
7110 PX=D
7120 RETURN
8000 REM  * * *  LILLIFOR'S TEST FOR NORMALITY  * *
8010 J=IO
8020 CLS:PRINT "SELECT OPTION":PRINT:PRINT
8030 PRINT "(1)   MEAN and VARIANCE are unknown":PRINT
8040 PRINT "(2)   MEAN is known, VARIANCE is unknown"
```

```
8050 KP=896:KS=2:GOSUB 140
8060 CLS:PRINT@10, "LILLIFOR'S TEST FOR NORMALITY"
8070 PRINT:PRINT
8080 INPUT "ENTER SKolmogorov Statistic";D
8090 IF D<0 OR D>1 THEN PRINT "* * ERROR * *":GOTO 8080
8100 INPUT "ENTER Sample Size";N
8110 IF N<1 THEN PRINT "* * ERROR * *  SAMPLE SIZE MUST BE
POSITIVE":GOTO 8100
8120 DENOM=1.0:PX=D
8130 D=PX/DENOM
8140 IF IO=1 THEN S=D*(-.01+SQR(N)+0.85/SQR(N)) ELSE
S=D*(0.055+SQR(N)+.283/SQR(N))
8150 TDENOM=DENOM
8160 GOSUB 8110
8170 IF IO>1 THEN GOTO 8260
8180 IF FOX <= .85 THEN DENOM=1-.24* FOX:GOTO 8230
8190 IF FOX <= .90 THEN DENOM=1.357- .66* FOX:GOTO 8230
8200 IF FOX <= .95 THEN DENOM = 1.339 - .64* FOX : GOTO
8230
8210 IF FOX <= .975 THEN DENOM = 1.795 - 1.12* FOX : GOTO
8230
8220 DENOM = 2.133-1.467* FOX
8230 IF ABS(TDENOM-DENOM) > .001 THEN GOTO 8130
8240 IO=J
8250 RETURN
8260 IF FOX <= .9 THEN DENOM=1.83-.8* FOX :GOTO 8230
8270 IF FOX <= .95 THEN DENOM =1.47 - .4* FOX : GOTO 8230
8280 IF FOX <= .975 THEN DENOM = 1.85 - .8* FOX : GOTO 8230
8290 DENOM = 1.07 : GOTO 8230
9000 REM  * * *  LILLIFOR'S TEST FOR EXPONENTIALITY  * * *
9010 CLS:PRINT@10, "LILLIFOR'S TEST FOR EXPONENTIALITY"
9020 PRINT:PRINT
9030 INPUT "ENTER SKolmogorov Statistic";D
9040 IF D<0 OR D>1 THEN PRINT "* * ERROR * *":GOTO 8080
9050 INPUT "ENTER Sample Size";N
9060 IF N<1 THEN PRINT "* * ERROR * *  SAMPLE SIZE MUST BE
POSITIVE":GOTO 8100
9070 IF D > 1.3442/(SQR(N)+.26+.5/SQR(N))+.2/N THEN PR=1 :
GOTO 9160
9080 PR=.5:GOSUB 19090
9090 IF D>XQ THEN P1=PR:P2=1:GOTO 9110
9100 P1=0:P2=PR
9110 PR=(P1+P2)/2
9120 GOSUB 19090
9130 IF ABS(D-XQ)<.0001 THEN GOTO 9160
9140 IF D>XQ THEN P1=PR ELSE P2=PR
9150 GOTO 9110
9160 FOX=PR
9170 QOX=1-FOX
9180 PX = D
9190 RETURN
```

```
10010 IF I2=1 GOTO 10080
10020 CLS: PRINT @20,"CHI-SQUARE DISTRIBUTION"
10030 PRINT @192,"";:INPUT"Enter Chi-Square Statistic  x
";CX
10040 IF CX<=0 PRINT"** ERROR **   CHI-SQUARE STATISTIC
MUST BE > 0":GOTO 10030
10050 IF I2=0 THEN PX=CX
10060 INPUT"Enter Degrees of Freedom";DF
10070 IF DF<=0 OR DF<>INT(DF) THEN PRINT"** ERROR **
ENTER AN INTEGER > 0":GOTO 10060
10080 IF DF>30 GOTO 10290
10090 DP=1
10100 FOR I=INT(DF) TO 2 STEP -2
10110 DP=DP
10120 NEXT I
10130 NP=CX[(INT((DF+1)/2))*EXP(-CX/2)/DP
10140 IF INT(DF/2)=DF/2 GOTO 10170
10150 PC=SQR(2/CX/3.141593)
10160 GOTO 10180
10170 PC=1
10180 LC=1
10190 MC=1
10200 D=DF
10210 D=D+2
10220 MC=MC* CX/D
10230 IF MC<.000001 GOTO 10260
10240 LC=LC+MC
10250 GOTO 10210
10260 FOX=PC*NP* LC
10270 QOX=1-FOX
10280 RETURN
10290 NX=((CX/DF)[(1/3) - (1-(2/(9*DF))))/SQR(2/(9*DF))
10300 MU=0 : VAR=1 : I2=1
10310 GOSUB 2070
10320 RETURN
13000 REM  * * *   INVERSE NORMAL DISTRIBUTION  * *
13010 IF I2=1 GOTO 13100
13020 CLS : PRINT @20, "INVERSE NORMAL"
13030 PRINT : PRINT : INPUT"Enter Mean";MO
13040 INPUT "Enter Variance";VR
13050 IF VR<=0 PRINT"** ERROR **   ";VR;"IS NOT A VALID
VARIANCE":GOTO 13040
13060 INPUT"Enter Probability (i.e.  F(x) =  Prob. ( X <= x
) = ";PR
13070 IF PR=0 THEN XQ=9E-36 ELSE IF PR=1 XQ=9E+36
13080 IF PR=0 OR PR=1 RETURN
13090 IF PR<0 OR PR>1 THEN PRINT"** ERROR **   ";PR;"IS NOT
A VALID PROBABILITY":GOTO 13060
13100 IF PR>.5 THEN RL=SQR(LOG(1/((1-PR)[2)))  ELSE
RL=SQR(LOG( 1/(PR[2)))
13110
XQ=RL-((2.515517+(.802853*RL)+(.010328*RL[2))/(1+(1.432788*R
```

```
L)+(.189269*RL[2)+(.001308*RL[3)))
13120 IF PR>=.5 THEN XQ=MO+(SQR(VR)*XQ) ELSE
XQ=MO-(SQR(VR)*XQ)
13130 RETURN
14000 REM * * *  INVERSE WILCOXSON SIGNED-RANKS * *
14010 CLS:PRINT "INVERSE WILCOXSON SIGNED-RANKS
DISTRIBUTION"
14020 PRINT:PRINT
14030 INPUT "ENTER Sample Size";IN
14040 IF IN < 1 THEN PRINT "* * *  ERROR * * * SAMPLE SIZE
MUST BE POSITIVE":GOTO 14030
14050 MO=IN*(IN+1)/4-.5
14060 VR=(2N+1)*(IN+1)N/24
14070 GOSUB 13060
14071 V=INT(XQ)
14072 GOSUB 3090
14073 TFOX=FOX
14074 V=V+1
14075 GOSUB 3090
14076 XQ=INT(XQ)+(CDBL(PR)-TFOX)/(FOX-CDBL(TFOX))
14080 MO=XQ-INT(XQ)
14090 IF MO<.26 THEN XQ=INT(XQ):RETURN
14100 IF MO<.75 THEN XQ=INT(XQ)+.5 ELSE XQ=INT(XQ)+1
14110 RETURN
15000 REM * * *  INVERSE MANN-WHITNEY  * *
15010 CLS:PRINT "INVERSE MANN-WHITNEY DISTRIBUTION"
15020 PRINT:PRINT
15030 INPUT "ENTER Smaller Sample Size";N
15040 IF N<1 THEN PRINT "* * * ERROR * * * SAMPLE SIZE MUST
BE POSITIVE":GOTO 15030
15050 INPUT "Enter Larger Sample Size";M
15060 IF M<1 THEN PRINT "* * * ERROR * * * SAMPLE SIZE MUST
BE POSITIVE":GOTO 15050
15070 MO=N*M/2
15080 VR=N*M*(N+M+1)/12
15090 GOSUB 13060
15091 U=INT(XQ)
15092 GOSUB 5130
15093 TFOX=FOX
15094 U=U+1
15096 GOSUB 5130
15097 XQ=INT(XQ)+(CDBL(PR)-CDBL(TFOX))/(FOX-CDBL(TFOX))
15100 MO=XQ-INT(XQ)
15110 IF MO<.26 THEN XQ=INT(XQ):RETURN
15120 IF MO<.75 THEN XQ=INT(XQ)+.5 ELSE XQ=INT(XQ)+1
15130 RETURN
16000 REM * * *  INVERSE KOLMOGOROV * *
16010 CLS : PRINT @20, "INVERSE KOLMOGOROV"
16020 PRINT:PRINT
16030 INPUT"ENTER Sample Size";N
16040 IF N<1 THEN PRINT "* * ERROR * *":GOTO 16030
16050 INPUT"Enter Probability (i.e.  F(x) =  Prob. ( X <= x
```

```
) = ";PR
16060 IF PR=1 THEN XQ= 0.0 ELSE IF PR=0 XQ=9E+36
16070 IF PR=0 OR PR=1 RETURN
16080 IF PR<0 OR PR>1 THEN PRINT"** ERROR **   ";PR;"IS NOT
A VALID PROBABILITY":GOTO 16050
16090 S=SQR(LOG(1-PR)/LOG(.135))
16100 D=S/(.12+SQR(N)+.11/SQR(N))
16110 XQ=D
16120 RETURN
17000 REM * * *  INVERSE SMIRNOV * *
17010 CLS : PRINT @20, "INVERSE SMIRNOV"
17020 PRINT:PRINT
17030 INPUT"ENTER Smaller Sample Size";N
17040 IF N<1 THEN PRINT "* * ERROR * *":GOTO 17030
17050 INPUT"ENTER Larger Sample Size";M
17060 IF M<N THEN PRINT "* * ERROR * *":GOTO 17030
17070 INPUT"Enter Probability (i.e.  F(x) =  Prob. ( X <= x
) = ";PR
17080 IF PR=1 THEN XQ= 0.0  ELSE IF PR=0 THEN XQ=9E+36
17090 IF PR=0 OR PR=1 RETURN
17100 IF PR<0 OR PR>1 THEN PRINT"** ERROR **   ";PR;"IS NOT
A VALID PROBABILITY":GOTO 17060
17110 S=SQR(LOG(1-PR)/LOG(.135))
17120 IF N<=0.1*M THEN D=S/(.12+SQR(N)+.11/SQR(N))+1/(2*M)
    ELSE IF (M/N)=INT(M/N) THEN
              D=(S-2/(3* SQR(M)))/SQR(N*M/(N+M))
    ELSE D=(S-2/(5* SQR(M)))/SQR(N*M/(N+M))
17130 XQ=D
17140 RETURN
18000 REM * * *  INVERSE LILLIEFOR'S NORMALALITY  * *
18010 J=IO:CLS:PRINT "SELECT OPTION":PRINT:PRINT
18020 PRINT "(1)   MEAN and VARIANCE are unknown":PRINT
18030 PRINT "(2)   MEAN is known, VARIANCE is unknown"
18040 KP=896:KS=2:GOSUB 140
18050 CLS:PRINT @13, "INVERSE LILLIEFOR'S NORMALITY TEST
DISTRIBUTION"
18060 PRINT:PRINT
18070 INPUT"ENTER Sample Size";N
18080 IF N<1 THEN PRINT "* * ERROR * *":GOTO 16030
18090 INPUT"Enter Probability (i.e.  F(x) =  Prob. ( X <= x
) = ";PR
18100 IF PR=0 THEN XQ=9E-36 ELSE IF PR=1 XQ=9E+36
18110 IF PR=0 OR PR=1 THEN IO=J:RETURN
18120 IF PR<0 OR PR>1 THEN PRINT"** ERROR **   ";PR;"IS NOT
A VALID PROBABILITY":GOTO 13060
18130 S=SQR(LOG((1-PR))/LOG(.135))
18140 IF IO=1 THEN D=S/(-.01+SQR(N)+.85/SQR(N)) ELSE
D=S/(.055+.283/SQR(N)+SQR(N))
18150 IF IO>1 THEN GOTO 18240
18160 IF PR <= .85 THEN DENOM=1-.24* FOX:GOTO 18210
18170 IF PR <= .90 THEN DENOM=1.357- .66*PR:GOTO 18210
18180 IF PR <= .95 THEN DENOM = 1.339 - 0.64*PR : GOTO
```

```
18210
18190 IF PR <= .975 THEN DENOM = 1.795 - 1.12*PR : GOTO
18210
18200 DENOM = 2.133-1.467*PR
18210 XQ=D*DENOM
18220 IO = J
18230 RETURN
18240 IF PR <= .9 THEN DENOM = 1.83 - .8*PR : GOTO 18210
18250 IF PR <= .95 THEN DENOM = 1.47 - .4*PR : GOTO 18210
18260 IF PR <= .975 THEN DENOM = 1.85 - .8*PR : GOTO 18210
18270 DENOM = 1.07:GOTO 18210
19000 REM * * *  INVERSE LILLIEFOR EXPONENTIAL TEST * *
19010 CLS:PRINT @13, "INVERSE LILLIEFOR'S EXPONENTIAL TEST
DISTRIBUTION"
19020 PRINT:PRINT
19030 INPUT"ENTER Sample Size";N
19040 IF N<1 THEN PRINT "* * ERROR * *":GOTO 16030
19050 INPUT"Enter Probability (i.e.  F(x) =  Prob. ( X <= x
) = ";PR
19060 IF PR=0 THEN XQ=9E-36 ELSE IF PR=1 XQ=9E+36
19070 IF PR=0 OR PR=1 RETURN
19080 IF PR<0 OR PR>1 THEN PRINT"** ERROR **   ";PR;"IS NOT
A VALID PROBABILITY":GOTO 19050
19090 DENOM=SQR(N)+.26+.5/SQR(N)
19100 IF PR > .95 THEN XQ = 5.125*PR - 3.7808 : GOTO 19190
19110 IF PR => .90 THEN XQ = 2.026*PR - .8367 : GOTO 19190
19120 IF PR => .80 THEN XQ = 1.124*PR - .0249 : GOTO 19190
19130 IF PR => .7 THEN XQ = .743*PR + .2799 : GOTO 19190
19140 IF PR => .5 THEN XQ = .546*PR  + .4187 : GOTO 19190
19150 IF PR => .3 THEN XQ = .464*PR + .4588 : GOTO 19190
19160 IF PR => .2 THEN XQ = .485*PR + .4525 : GOTO 19190
19170 IF PR => .1 THEN XQ = .587*PR + .4321 : GOTO 19190
19180 XQ = .846*PR + .4062
19190 XQ=XQ/DENOM
19200 XQ = XQ + .2/N
19210 RETURN
19220 REM
19230 IF I2=1 GOTO 20070
20000 CLS : PRINT @20,"INVERSE CHI-SQUARE"
20010 PRINT : INPUT"Enter Degrees of Freedom";DF
20020 IF DF<=0 OR DF<>INT(DF) THEN PRINT"** ERROR **
ENTER AN INTEGER > 0":GOTO 20010
20030 INPUT "Enter Probability   i.e. F(x) = Prob. ( X <= x
) = ";PR
20040 IF PR=0 THEN XQ=0 : RETURN
20050 IF PR=1 THEN XQ=9E36 : RETURN
20060 IF PR<0 OR PR>1 THEN PRINT"** ERROR **   ";PR;"IS NOT
A VALID PROBABILITY":GOTO 20030
20070 MO=0 : VR=1 : I2=1 : PZ=PR
20080 IF DF>1 GOTO 20130
20090 PR=.5*(1-PZ)
20100 GOSUB 13100
```

```
20110 XQ=XQ[2 : PR=PZ
20120 RETURN
20130 IF DF>2 GOTO 20160
20140 XQ=-2* LOG(1-PZ)
20150 RETURN
20160 GOSUB 13100
20170 DQ=XQ
20180 IF DF>(2+INT(4*ABS(DQ))) GOTO 20280
20190 X9=1.0000886 - .2237368/DF - .01513904/DF[2
20200 X9=X9+(DF[-.5 * DQ * (.4713941 + .02607083/DF -
.008986007/DF[2))
20210 X9=X9 + (1/DF * DQ[2 * (.0001348028 + .01128186/DF +
.02277679/DF[2))
20220 X9=X9 + (DF[-1.5 * DQ[3 * (-.008553069 - .01153761/DF
- .01323293/DF[2))
20230 X9=X9 + (DF[-2 * DQ[4 * (.00312558 + .005169654/DF -
.006950356/DF[2))
20240 X9=X9 + (DF[-2.5 * DQ[5 * (-.0008426812 +
.00253001/DF + .001060438/DF[2))
20250 X9=X9 + (DF[-3 * DQ[6 * (.00009780499 - .001450117/DF
+ .001565326/DF[2))
20260 XQ=DF*X9[3
20270 RETURN
20280 X9=1 - 2/(9*DF) + (4*DQ[4 + 16*DQ[2 -28)/(1215*DF[2)
20290 X9=X9 + (8*DQ[6 + 720*DQ[4 + 3126*DQ[2 +
2904)/(229635*DF[3)
20300 X8=DQ/3 + (-DQ[3 + 3*DQ)/(162*DF)
20310 X8=X8 - (3*DQ[5 + 40*DQ[3 + 45*DQ)/(5832*DF[2)
20320 X8=X8 + (301*DQ[7 - 1517*DQ[5 -32769*DQ[3 -
79349*DQ)/(7873200*DF[3)
20330 X8= SQR(2/DF) * X8
20340 XQ=DF * ((X8+X9)[3)
20350 RETURN
```

LIST OF REFERENCES

1. Claypool, P.L. and Holbert, D., Accuracy of Normal and Edgeworth Approximations to the Distributions the Wilcoxson Signed Rank Statistic, Journal of the American Statistical Association, March 1974, Volume 69:255-258

2. Chow, W.K. and Hodges, J.L. Jr., An Approximation for the Distribution of the Wilcoxson One-Sample Statistic, Journal of the American Statistical Association, September 1975, Volume 70:648-655

3. Fellingham, S.A. and Stoker, D.J., An Approximation for the Exact Distribution of the Wilcoxson Test for Symmetry, Journal of the American Statistical Association, September 1964, Volume 59:899-905

4. Isbell, R.P., Statistical Analysis Software for the TRS-80 Microcomputer, Masters Thesis, Naval Postgraduate School, September 1981

5. Fix, E. and Hodges, J.L. Jr., Significance Probabilities of the Wilcoxson Test, Annals of Mathematical Statistics, 1955, Volume 26:301-312

6. Drion, E.F., Some Distribution-Free Tests for the Difference Between Two Empirical Cummulative Distribution Functions, 1952, Annals of Mathematical Statistics, Volume 23, 563-574

7. Kim, P.J., On the Exact and Approximate Sampling Distribution of the Two Sample Kolmogorov-Smirnov Criterion, Journal of the American Statistical Association, December 1969, Volume 64:1625-1635

8. Lilliefors, H.W., On the Kolmogorov-Smirnov Test for Normality With Mean and Variance Unknown, Journal of the American Statistical Association, September 1967, Volume 62:399-402

9. Stephens, M.A., EDF Statistics for Goodness of Fit and Some Comparisons, Journal of the American Statistial Association, September 1974, Volume 69:730-737

10. Lilliefors, H.W., On the Kolmogorov-Smirnov Test for the Exponential Distribution With the Mean Unknown, Journal of the American Statistical Association, March 1969, Volume 64:387-389

11. Box,G., Hunter, G. and Hunter, J., Statistics for Experimenters, John Wiley & Sons, 1978, Page 150

BIBLIOGRAPHY

Conover, W.J., Practical Non-Parametric Statistics, Wiley, 1980

Lehmann, E.L., Non-Parametric Statistical Methods, Based on Ranks, McGraw-Hill, 1975

Tate, M.W. and Clelland, R.C., Non-Parametric Statistics, Interstate Pinters and Publishers, 1959

Wilcoxson, F., Individual Comparisons by Ranking Methods, Biometrics, 1945, Volume 1:80-88

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, CA 93940	1
4.	Professor J.D. Esary, Code 55EY Department of Operations Research Naval Postgraduate School Monterey, CA 93940	1
5.	Professor R.R. Read, Code 55RE Department of Operations Research Naval Postgraduate School Monterey, CA 93940	1
6.	Robert L. Zangmeister, Jr. Naval Ship Weapons Systems Engineering Station, Code 6221 Port Hueneme, CA 93043	1
7.	Maj. R.P. Isbell, USMC Marine Corps Tactical Systems Support Activity (MCTSSA) Marine Corps Base, Camp Pendleton California 92055	1

200155

Thesis
Z2423 Zangmeister
c.1 Non-parametric statistical software for the TSR-80 micro-computer.

22 APR 86 30347

200155

Thesis
Z2423 Zangmeister
c.1 Non-parametric statistical software for the TSR-80 micro-computer.

thesZ2423
Non-parametric statistical software for
3 2768 000 98779 6
DUDLEY KNOX LIBRARY